A0597203 AI Business Applications

Introduction to Generative AI (GenAI)

https://www.knime.com/events/genai-data-workflows-getting-started-course

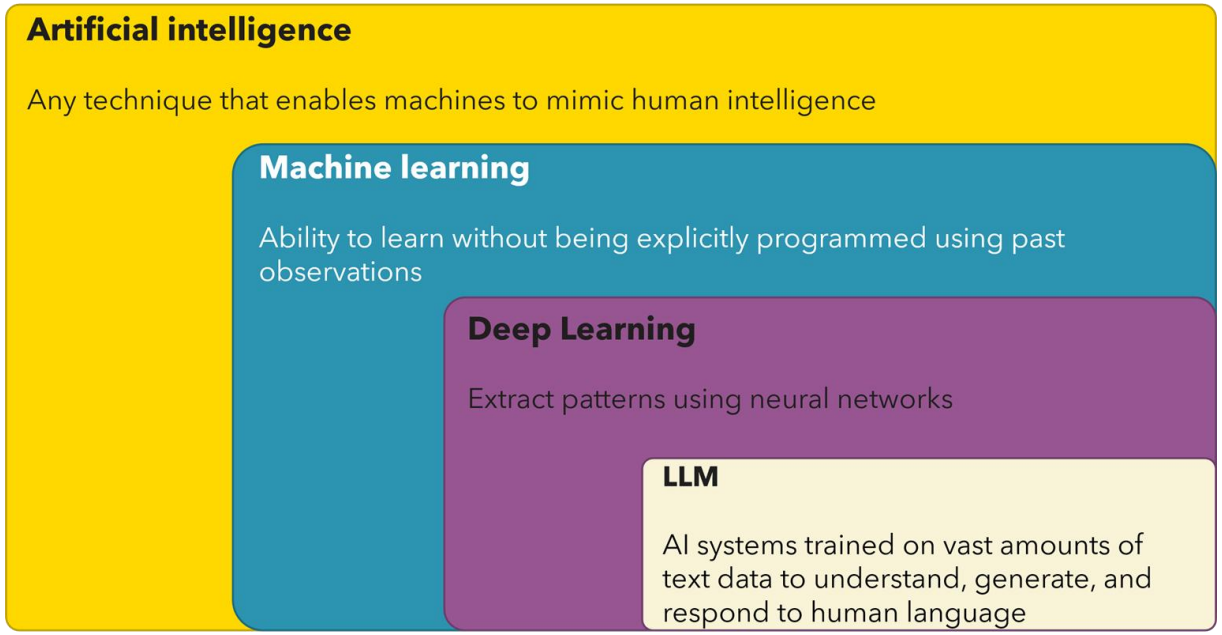
Introduction to Generative AI

The AI Landscape: A Quick Recap

In our previous sessions, we explored two major categories of artificial intelligence:

Supervised Learning (Predictive AI):

- This is like giving the AI a spreadsheet with questions and answers.
- The AI learns to predict the answer to a new question based on the examples.
- Think of an AI that classifies customer emails as "urgent" or "not urgent" based on past, labeled examples.



Unsupervised Learning (Descriptive AI):

- Here, we give the AI a dataset without any labels or answers.
- The AI's job is to find hidden patterns and relationships on its own.
- A good example is an AI that groups customers into different segments based on their purchasing behavior, helping a business understand its market better.

These two types of AI are primarily focused on analyzing and understanding existing data. They classify, predict, or find patterns.

What is Generative AI?

Generative AI is a different class of AI.

Instead of just analyzing data, it's designed to **create new, original content** that has never existed before. This content can take many forms:

- **Text:** Articles, emails, code, poems, scripts, and marketing copy.
- **Images:** Photos, digital art, logos, and product designs.
- **Audio:** Music, voiceovers, and sound effects.
- **Video:** Short clips, animations, and movie scenes.

Why is it called "Generative"?

- The term "generative" simply means "capable of producing or creating."
- The AI doesn't just retrieve information; it synthesizes it in new and imaginative ways to generate something novel.
- This is a massive leap forward, as it moves AI from a tool for analysis to a partner for **creation and innovation**.

Why Does Generative AI Matter for Business?

Generative AI is a powerful tool for driving business value. Its ability to create content can:

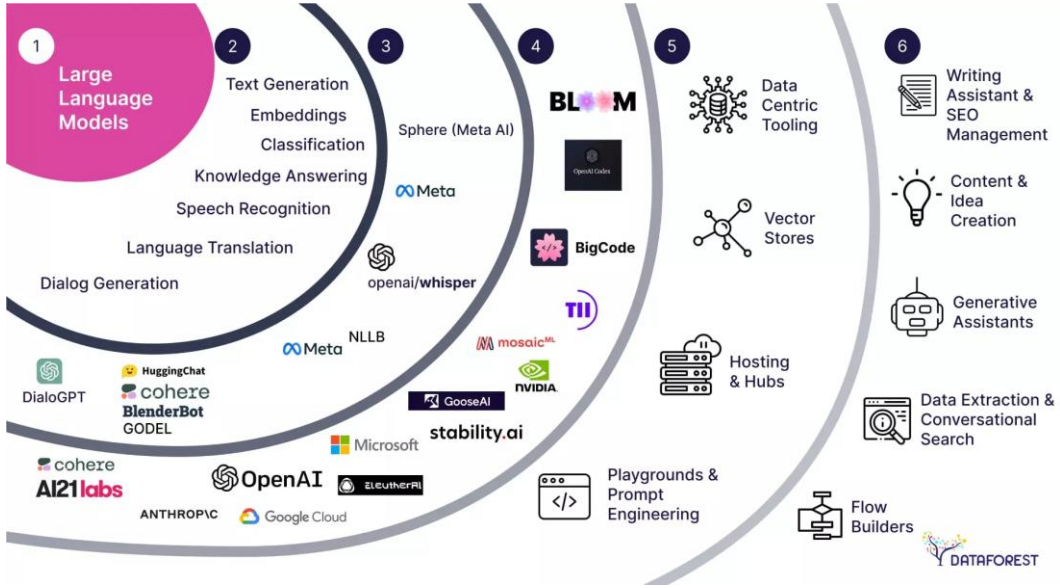
- **Business Applications**: It enhances customer service through AI-powered chatbots, automates report generation, and brings in optimizations in various processes in terms of cost and manual effort.
- **Increase Efficiency:** It can automate time-consuming, repetitive tasks like drafting emails, writing product descriptions, or summarizing documents, allowing employees to focus on higher-value work.
- **Unlock Creativity and Innovation:** It can act as a brainstorming partner, generating new ideas for products, marketing campaigns, or even business strategies. This can accelerate the creative process from days to minutes. Also, generative AI is used in art, music, and writing to create new pieces that mimic human creativity. Tools like DALL-E and ChatGPT are popular examples.
- **Personalize at Scale:** Businesses can use it to create highly personalized content for individual customers, such as customized marketing emails or product recommendations, at a massive scale that was previously impossible.

- In essence, generative AI is a new type of intelligent automation that can handle complex, creative tasks.
- For a business leader, understanding how to apply this technology is key to gaining a competitive advantage.

Large Language Models (LLMs)

Large Language Models (LLMs)

- **LLMs** are the **engines** that power popular tools like ChatGPT, Gemini, and Claude.
- Large Language Model is a massive **Neural Networks** that has been trained on an enormous amount of text and data from the internet—books, articles, websites, and more.
- This training process isn't about memorizing facts; it's about learning the **patterns, grammar, and relationships** within language.
- These models can understand and generate human language, enabling tasks like answering questions, summarizing text, or writing content.



1 — Available Large Language Models

2 — General Use-Cases

3 — Specific Implementations

4 — Models

5 — Foundation Tooling

6 — End User UIs

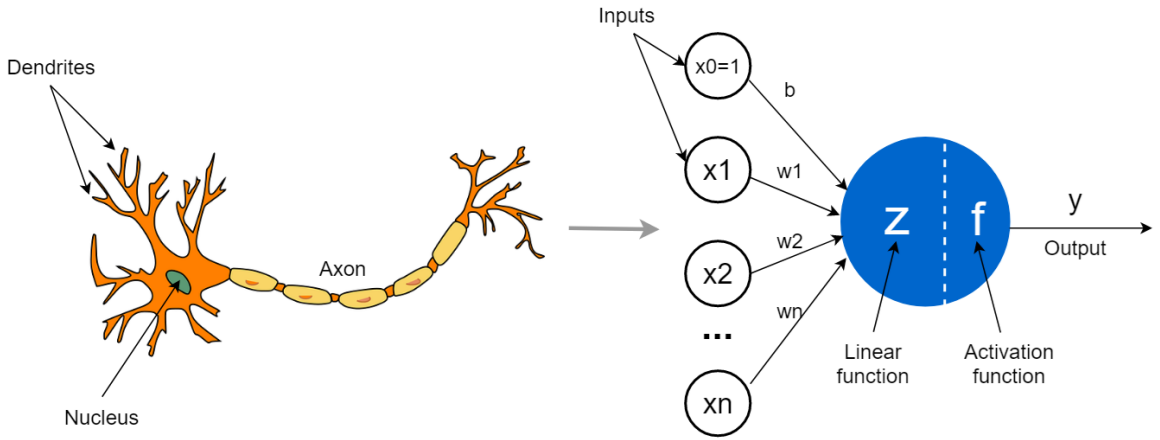
https://dataforest.ai/blog/large-language-models-advanced-communication

What is a Neural Network?

- To understand how LLMs work, it helps to understand some fundamentals of neural networks, as they are the foundation of many AI systems.
- Neural networks are inspired by the brain and consist of interconnected “neurons” that adjust connection strengths during learning.
- 1943 – McCulloch & Pitts: Proposed the first mathematical model of a neuron (MCP neuron), a binary threshold logic gate.
- They showed that networks of such neurons could compute any logical function.
- This laid the theoretical foundation for artificial neural networks.

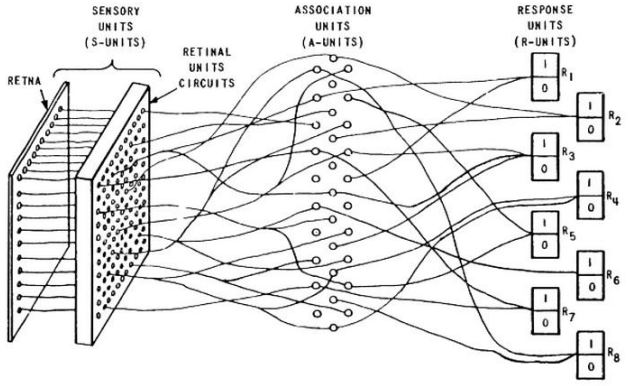
Warren Sturgis McCulloch
(1898 – 1969)

Walter Harry Pitts, Jr.
(1923 – 1969)

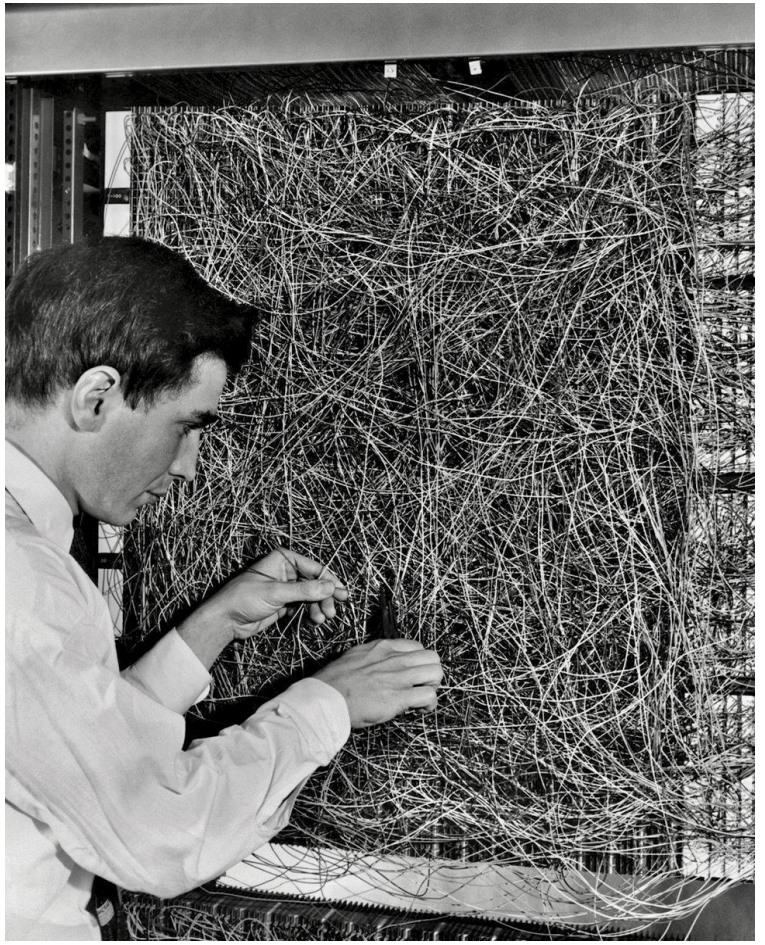


The Perceptron: Building Block of Neural Networks

- 1958 – Frank Rosenblatt (inspired by McCulloch & Pitts): Introduced the **Perceptron**.
- Designed to recognize patterns in visual data.
- Implemented in hardware (the Mark I Perceptron at Cornell).
- The perceptron is a binary classifier: input → weighted sum → threshold → output.
- Rosenblatt also described multi-layer perceptrons, but at the time no algorithm existed to train them effectively.



F. Rosenblatt

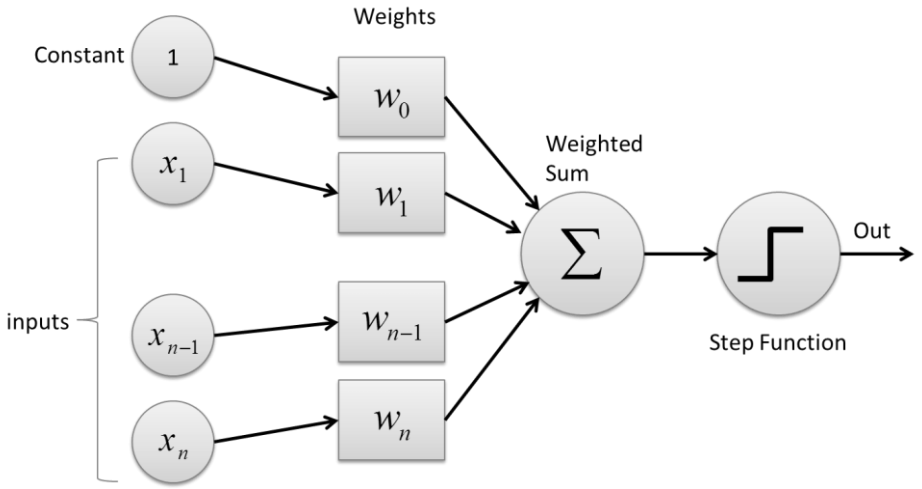


The diagram shows **Rosenblatt's Perceptron (1958)**, which is essentially an early form of a **multi-layer perceptron (MLP)**: sensory units (S-units) connected to association units (A-units), which in turn connect to response units (R-units). Inspired by the visual cortex, it could learn to classify input patterns by adjusting connection weights. While the theory was purely mathematical, Rosenblatt also built an **electromechanical implementation called the Mark I Perceptron**, which used an array of photocells as the retina, analog circuits for weighted connections, and motors to adjust the weights physically. This made it one of the first tangible demonstrations of machine learning hardware.

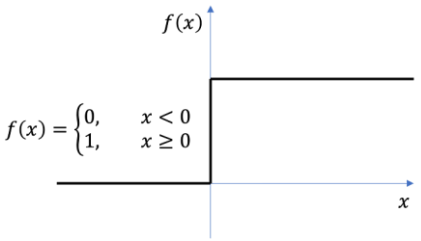
The Components of the Perceptron

1. Inputs: $x_1, x_2, \ldots, x_n$
2. Weights: $w_1, w_2, \ldots, w_n$
3. Bias: b
4. The Activation function

The original Perceptron used the step function which outputs 1 if $z \geq 0$ and outputs 0 if $z < 0$



Step Function:
Output: 1 if $z \geq 0$, 0 if $z < 0$
Used in original Perceptron
Not differentiable at 0



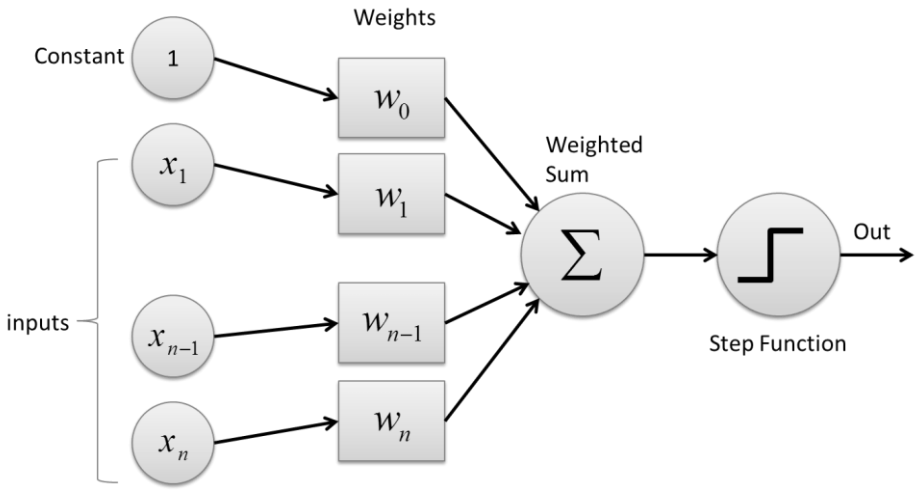
How Does the Perceptron Work?

1. Multiply each input by its corresponding weight
2. Sum all weighted inputs
3. Add the bias term
4. Apply the activation function (step function)
5. Output the result

Mathematically:

$z = w_1x_1 + w_2x_2 + ... + w_nx_n + b$

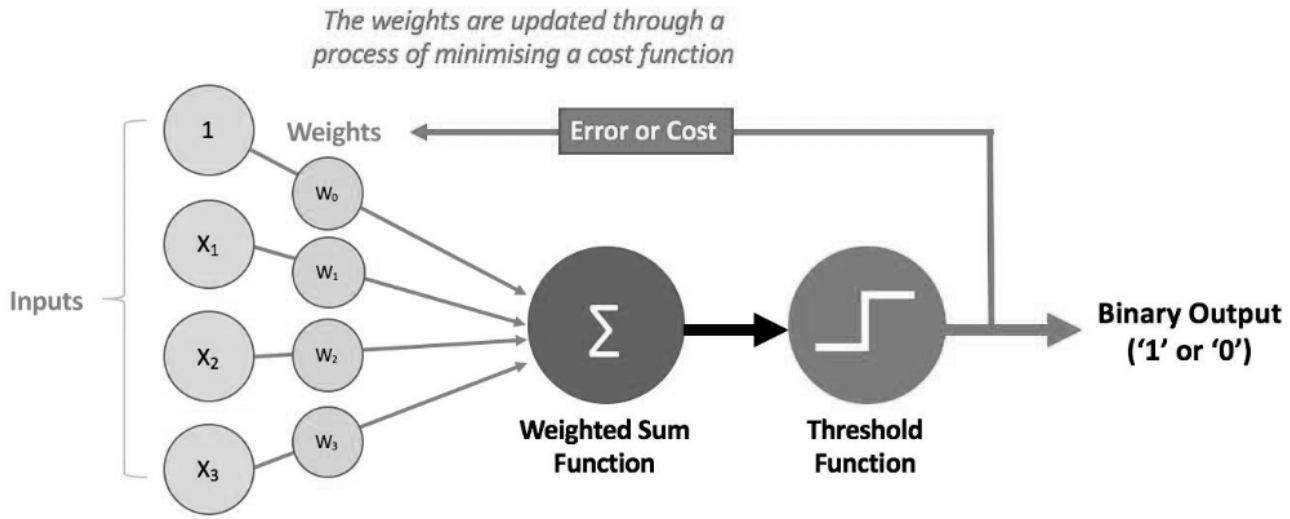
output = activation(z) 1 if $z \geq 0$, 0 if $z < 0$



How Does the Perceptron Learn?

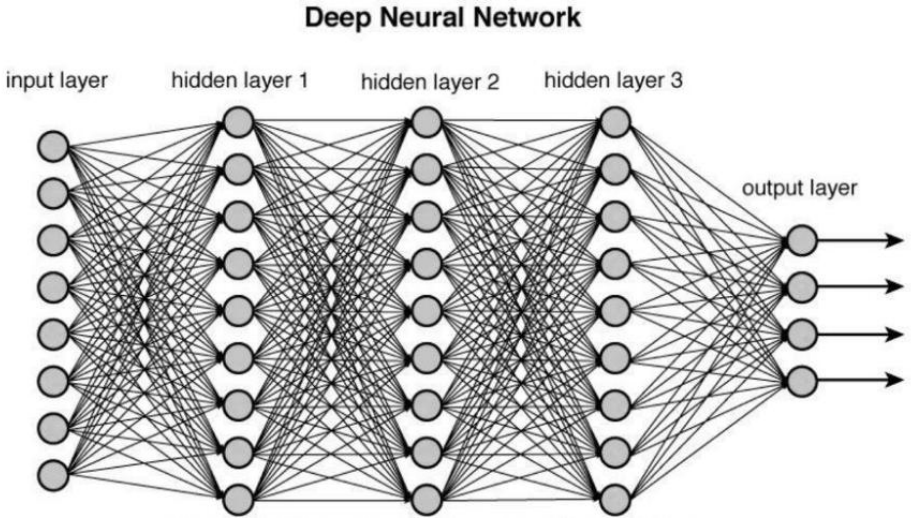
For each training example:

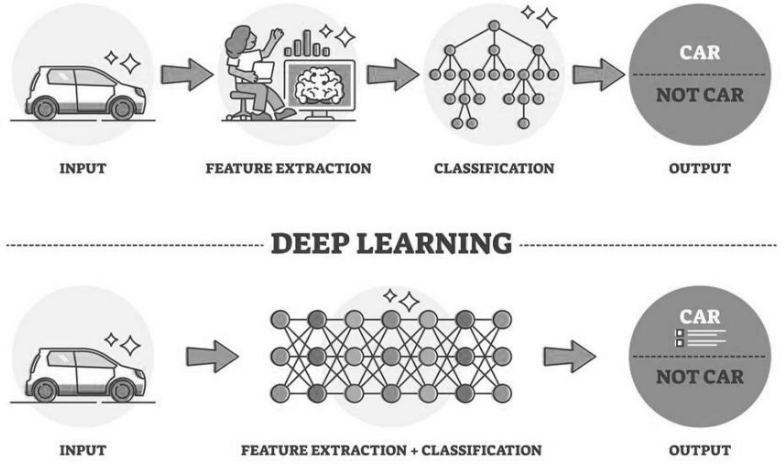
1. Calculate predicted output y_pred
2. Calculate error: error = y_true - y_pred
3. Update weights: w_new = w_old + learning_rate * error * x
4. Update bias: b_new = b_old + learning_rate * error



Deep Neural Networks and Deep Learning

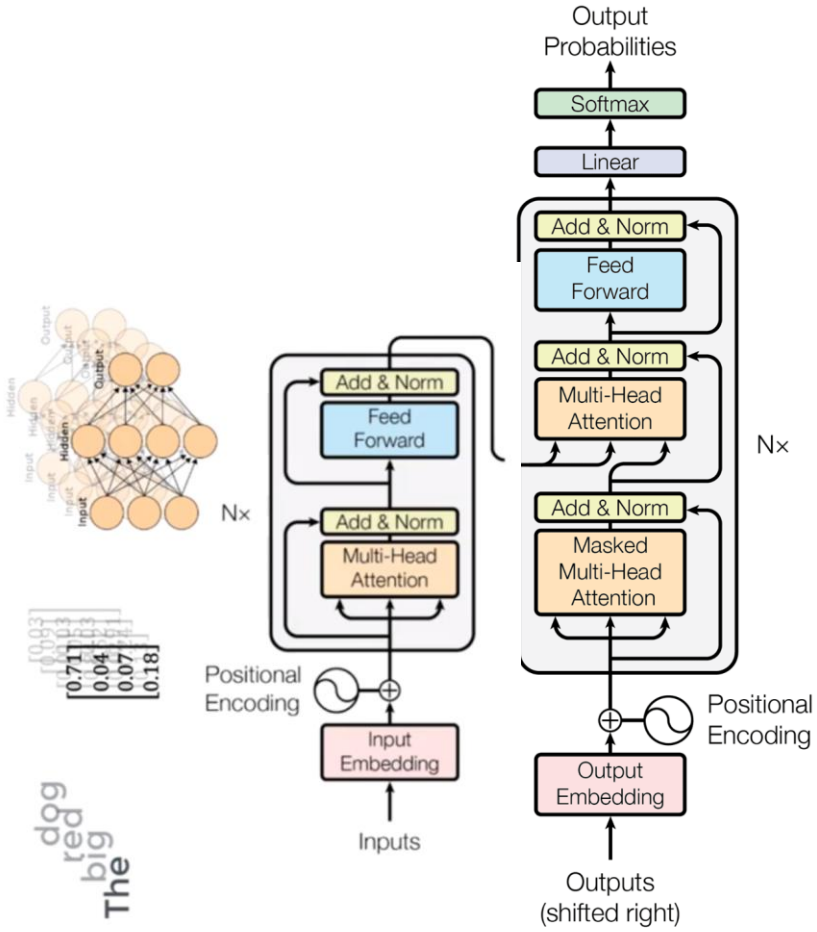
- Neural Networks have revolutionized artificial intelligence by enabling machines to learn from data in ways that mimic human neural processes.
- Deep Neural Networks (DNNs) are Neural Networks that are composed of multiple processing layers that can learn representations of data with multiple levels of abstraction.
- The power of deep learning comes from:
  - Its ability to automatically discover intricate patterns in raw data through the learning process, without requiring human engineers to manually specify all the knowledge needed by the computer system, therefore, constructing multiple levels of abstraction
  - Its ability to automatically extract patterns
  - Its scalability with big data and GPUs
- This foundation enabled modern AI systems, including Large Language Models (LLMs).





Large Language Models (LLMs)

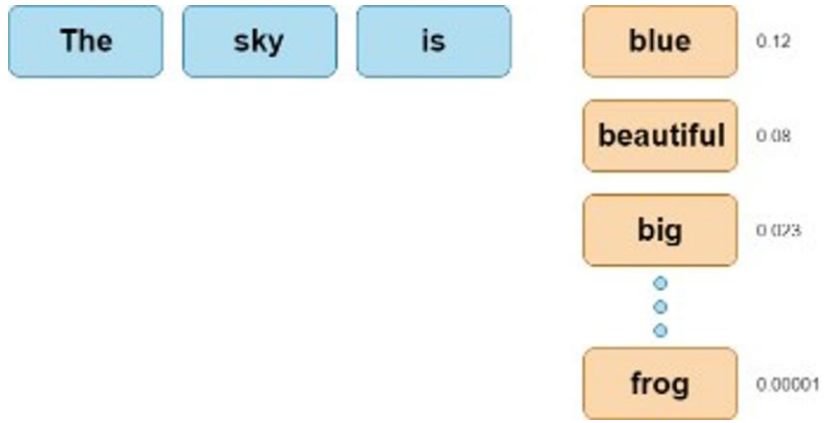
- Large Language Models are built on the Transformer Architecture, which is a neural network architecture that is well-suited for processing sequences.
- Transformer architecture is a LARGE Neural Networks that can reach thousands of neurons and billions of weights.
- These models are trained to learn patterns in human language, enabling them to generate coherent text and support more natural, intuitive interactions with technology.



How do LLMs work?

- LLMs function as complex auto-completion systems.
- They are designed to suggest the most likely next word based on their exposure to similar contexts during training.
- Consider this example:

  when given the phrase “The sky is”, an LLM predicts the most likely next word based on learned patterns.
- “Blue” might rank highest, while words like “beautiful” or “big” are also plausible.
- Less likely words, like “frog”, may still receive some probability, even if they don’t fit the context.



- LLMs operate based on probabilistic knowledge, predicting words and phrases according to patterns in data rather than true understanding.
- They do not possess a semantic understanding of the content they generate.

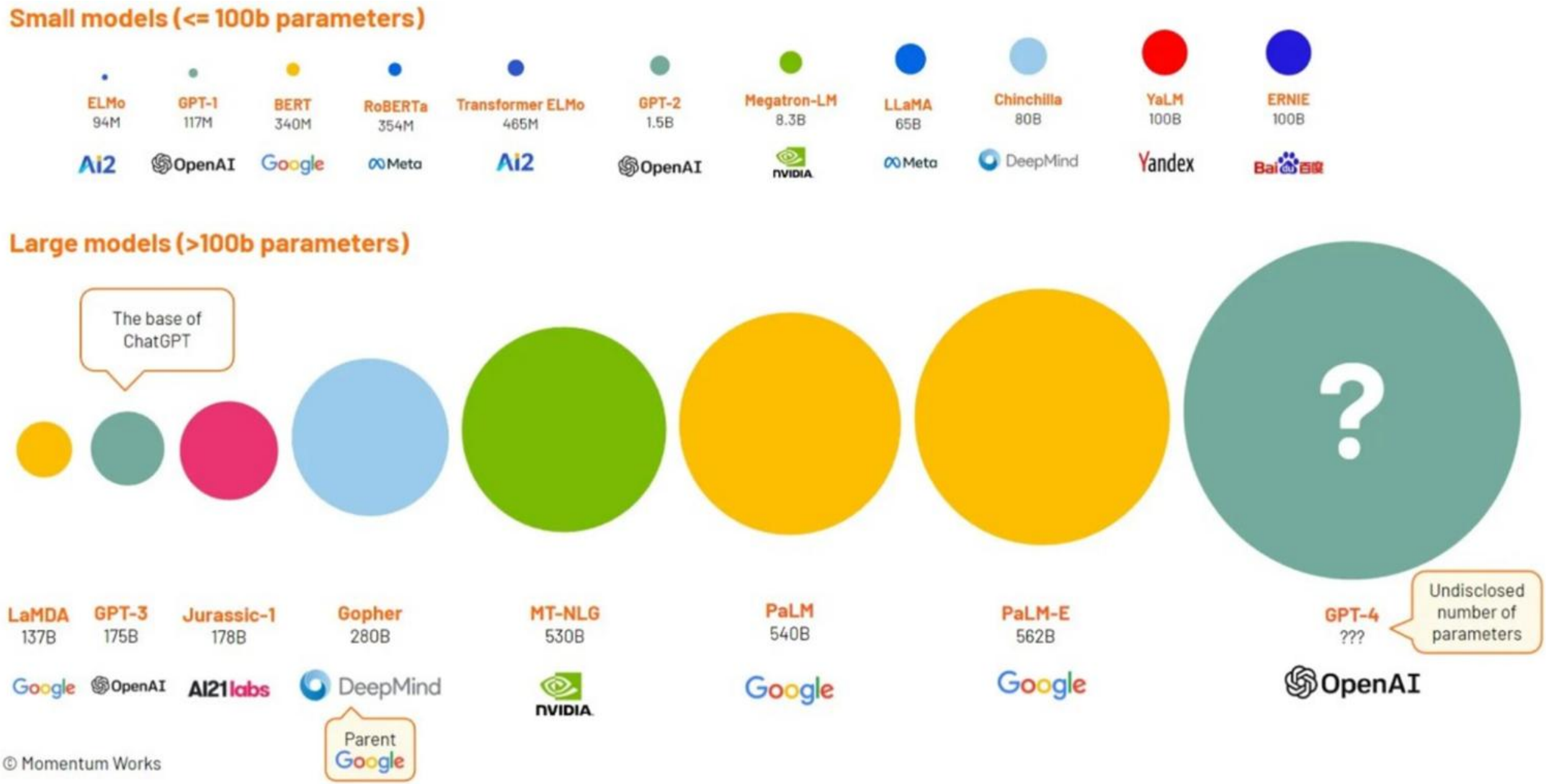
YouTube

Why are LLMs called large language models?

- LLMs are called "large" because of their immense scale in both architecture and training data.
- LLMs consist of billions to trillions of trainable parameters—internal values the model learns and adjusts to identify patterns and generate accurate outputs.
- In general, the larger the model, the better its performance on a wide range of language tasks.

> For example, OpenAI's GPT-3 has 175 billion parameters, while GPT-4 is estimated to have around 1.76 trillion, demonstrating how size correlates with capability.

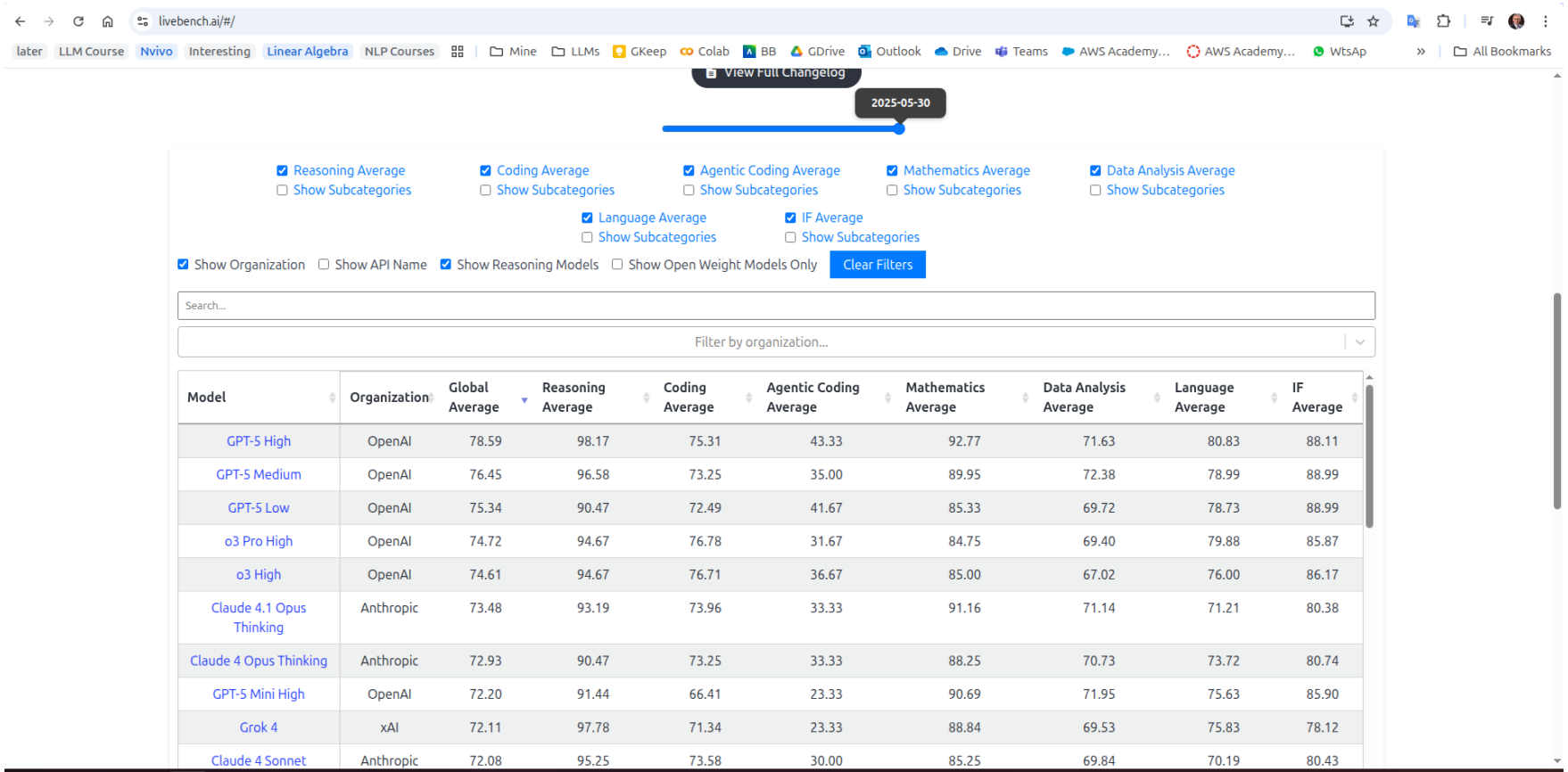
- In addition to their size, LLMs are trained on *massive datasets* and require *substantial computational resources*, making it impractical for most users to train them from scratch.
- This combination of vast parameter counts, massive training data, and advanced computing power is what makes these models truly large.



Source: https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/

Benchmarks for LLMs Performance

- LLM benchmarking is the process of using standardized datasets, tasks, and metrics to evaluate and compare the performance of large language models (LLMs) across various capabilities, such as language understanding, reasoning, coding, and factual recall.
- This helps developers identify an LLM's strengths and weaknesses, select the best model for a specific task, and guide efforts to improve its overall performance.
- LiveBench.ai is one of the available benchmarks that provides several information and comparisons between major LLMs.



Model	Organization	Global Average	Reasoning Average	Coding Average	Agentic Coding Average	Mathematics Average	Data Analysis Average	Language Average	IF Average
GPT-5 High	OpenAI	78.59	98.17	75.31	43.33	92.77	71.63	80.83	88.11
GPT-5 Medium	OpenAI	76.45	96.58	73.25	35.00	89.95	72.38	78.99	88.99
GPT-5 Low	OpenAI	75.34	90.47	72.49	41.67	85.33	69.72	78.73	88.99
o3 Pro High	OpenAI	74.72	94.67	76.78	31.67	84.75	69.40	79.88	85.87
o3 High	OpenAI	74.61	94.67	76.71	36.67	85.00	67.02	76.00	86.17
Claude 4.1 Opus Thinking	Anthropic	73.48	93.19	73.96	33.33	91.16	71.14	71.21	80.38
Claude 4 Opus Thinking	Anthropic	72.93	90.47	73.25	33.33	88.25	70.73	73.72	80.74
GPT-5 Mini High	OpenAI	72.20	91.44	66.41	23.33	90.69	71.95	75.63	85.90
Grok 4	xAI	72.11	97.78	71.34	23.33	88.84	69.53	75.83	78.12
Claude 4 Sonnet	Anthropic	72.08	95.25	73.58	30.00	85.25	69.84	70.19	80.43

The Three Ways to Make Use of LLMs

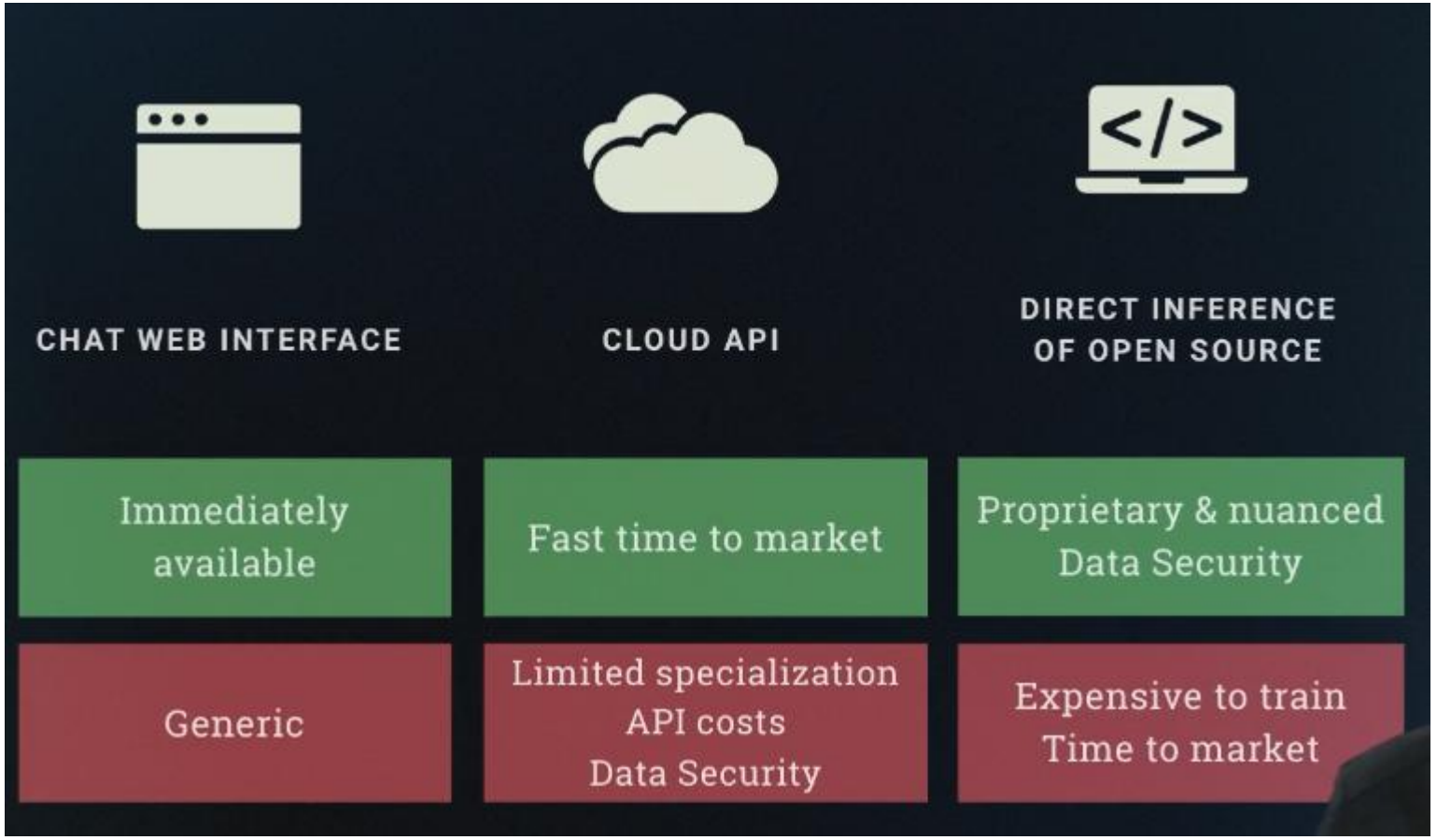


Image Generation - Diffusion Model

- **Definition**: Much like LLMs are the leading type of generative AI models for NLP tasks, diffusion models are the state-of-the-art approach for generating visual content like images and art.
- The principle behind diffusion models is to gradually add noise to an image and then learn to reverse this process through denoising.
- By doing so, the model learns highly intricate patterns, ultimately becoming capable of creating impressive images that often appear photorealistic.

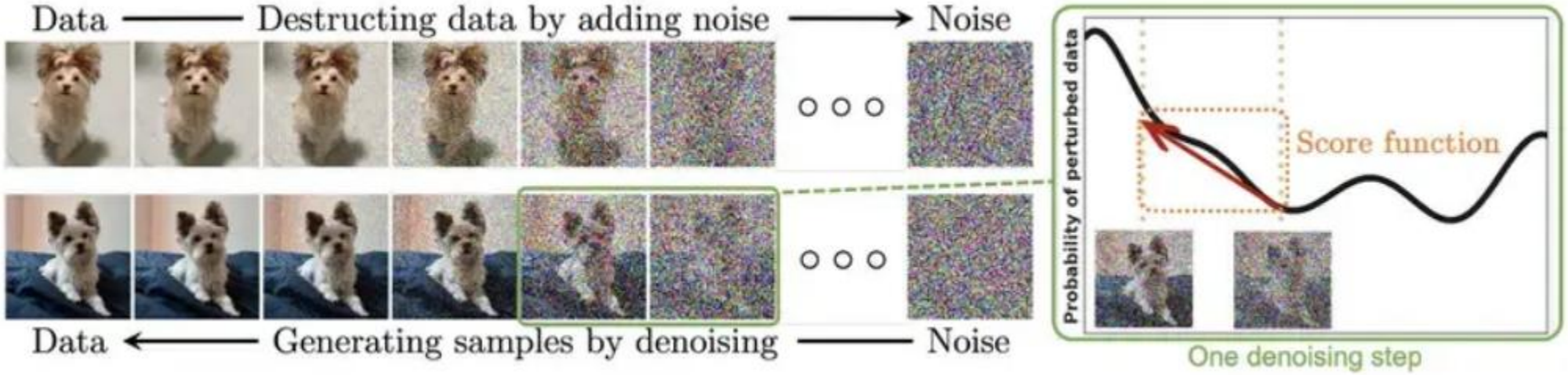


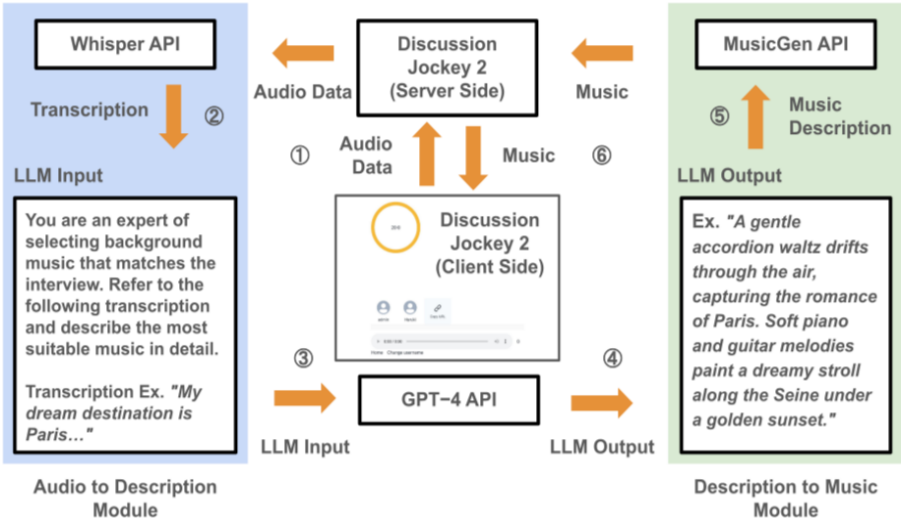
Image Generation

- Models like **DALL-E**, **Midjourney**, and **Stable Diffusion** are trained on a massive dataset of images and their corresponding text descriptions.
- When you give them a prompt like "a photorealistic image of a cat in a business suit drinking coffee," the model generates a new, original image that matches the description.
- **Business Applications:**
  - **Marketing & Advertising:** Quickly create unique visuals for social media campaigns, ad banners, or blog posts without a designer.
  - **Product Design:** Generate new product mockups or explore different design options in a fraction of the time.
  - **E-commerce:** Create lifelike product photos or virtual try-on experiences without expensive photo shoots.

1. https://openai.com/index/dall-e-2/
2. https://www.midjourney.com/home

Audio and Music Generation

- These models can create original music, sound effects, or lifelike human speech.
- They are trained on vast libraries of audio data and can generate new compositions based on a text description or a musical theme.
- **Business Applications:**
  - **Content Creation:** Generate royalty-free background music for a video or a podcast.
  - **Customer Experience:** Create natural-sounding voiceovers for virtual assistants or automated phone systems.
  - **Gaming:** Dynamically create sound effects and soundtracks that adapt to the game's environment.



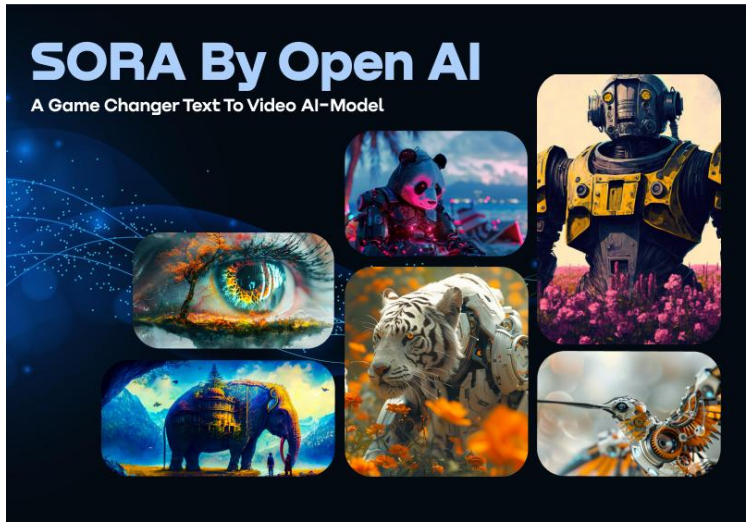
The diagram shows a workflow for automatically generating background music that matches spoken dialogue using AI models.

1. Whisper API transcribes spoken audio into text.
2. The transcription is passed into the GPT-4 API, which acts as an Audio-to-Description Module. It analyzes the context and creates a descriptive explanation of the most suitable music style.
3. This music description (e.g., “a gentle accordion waltz...”) is sent to MusicGen API, the Description-to-Music Module, which generates actual music.
4. The Discussion Jockey 2 (Server + Client Side) coordinates the flow of audio data, transcription, music description, and generated music.

Essentially, the system listens to speech, interprets the context, and then produces matching background music automatically.

Video Generation

- Models like **Sora** can be used to create high-quality, realistic video clips from simple text prompts.
- **Business Applications (Emerging):**
  - **Entertainment:** Rapidly prototype animated storyboards for films or create short marketing videos.
  - **Training:** Generate realistic simulations for employee training, such as safety or machinery operation videos.



1. https://sora.chatgpt.com/

Limitations and Ethical Considerations

Hallucinations:

1. Definition: Generative AI models can confidently produce false, nonsensical, or made-up information.
2. Reason: They are trained to generate plausible sequences of words, not necessarily to be factual.
3. Mitigation: Always verify information generated by AI, especially for critical business decisions.

Bias:

1. Source: AI models learn from the data they are trained on. If the training data contains societal biases (e.g., gender, racial), the AI can perpetuate or amplify them.
2. Implication: AI-generated content might reflect these biases, leading to unfair or inappropriate outputs.
3. Mitigation: Be aware of potential biases and critically review AI outputs for fairness and inclusivity.

Data Privacy and Security:

1. Concern: When you input sensitive company data into public AI models, there's a risk of that data being used for training or exposed.
2. Best Practice: Avoid inputting confidential or proprietary information into general-purpose AI tools.

Copyright and Intellectual Property:

1. Issue: Who owns the content generated by AI? Can AI generate content that infringes on existing copyrights?
2. Current State: This is an evolving legal area. Be mindful of these complexities, especially for commercial use.

KNIME for Generative AI

Getting Started

1. Download KNIME Analytics Platform

2. Install the KNIME AI Extension (Labs)—this gives you the nodes for local and remote LLM connections.

The KNIME AI extension provides dedicated nodes for connecting to LLMs and embedding models of both commercial and open-source providers; prompting and chatting with LLMs, creating and managing vector stores, as well as implementing your chatbots, RAG pipelines, and agents.

3. Choose a language model

4. Build your workflow: Select model → Prompt (via LLM Prompter or LLM Chat Prompter) → Process outputs

Key Nodes in the AI Extension

Authentication Nodes:

- **Credentials Configuration:** Stores API keys securely
- **OpenAI Authenticator:** Authenticates with OpenAI services
- **Azure OpenAI Authenticator:** For Microsoft Azure integration
- **HuggingFace Authenticator:** For Hugging Face Hub models

Model Connection Nodes:

- **OpenAI LLM Selector:** Establishes connection with OpenAI LLM, allowing selection from available models
- **GPT4All LLM Connector:** For local model integration
- **Anthropic LLM Selector:** For Claude models

Prompting Nodes:

- **LLM Prompter:** Sends simple text prompts to a language model for one-shot prompting
- **Chat Model Prompter:** For conversational interactions
- **Agent Prompter:** Allows creation of agents with underlying LLMs and specialized tools

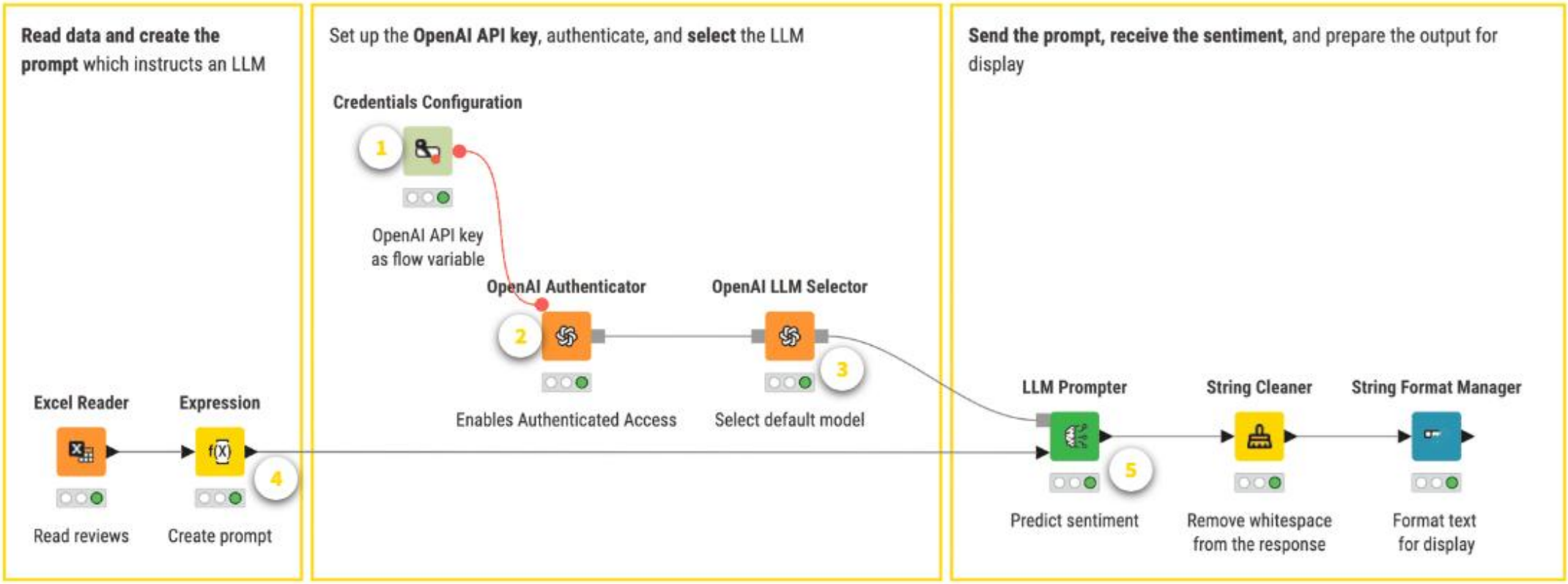
3 steps to leverage API-based LLMs

Independent of the provider, there are always 3 steps that you always need to perform:

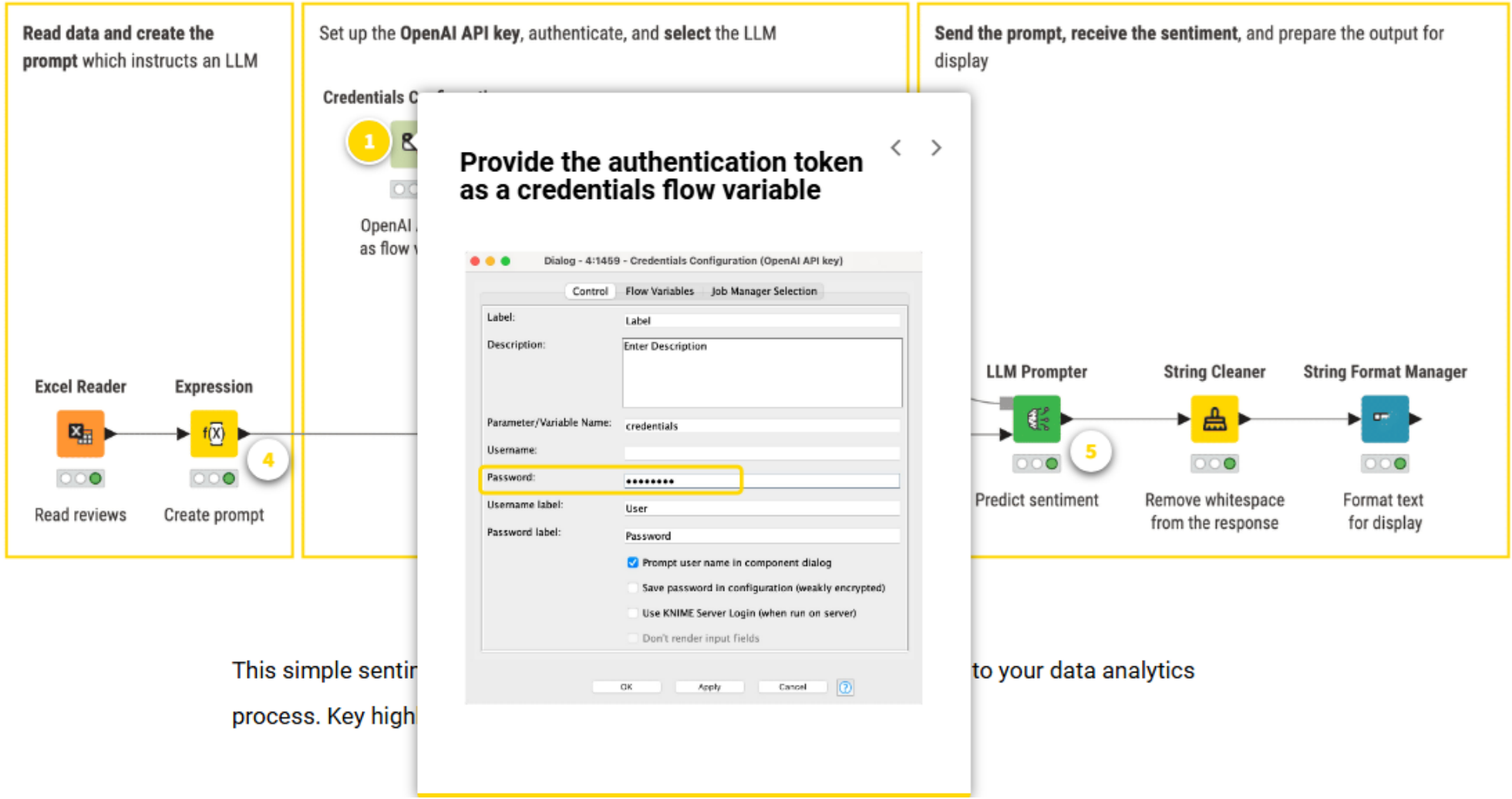
1. Authenticate against the provider & connect.
2. Select the model.
3. Prompt the model.

Sentiment Analysis Workflow

The diagram below shows a simple sentiment analysis workflow, where an LLM is used to evaluate the sentiment of customer reviews.

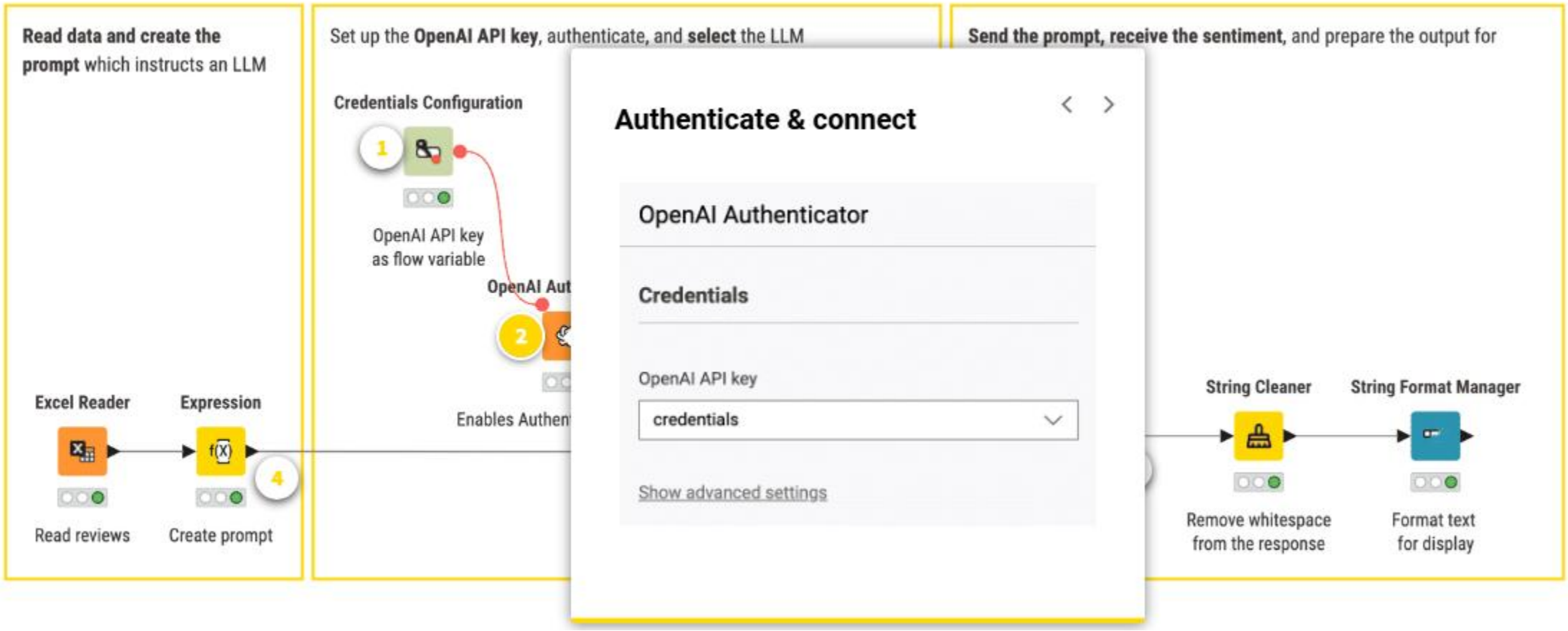


Provide Authentication Token

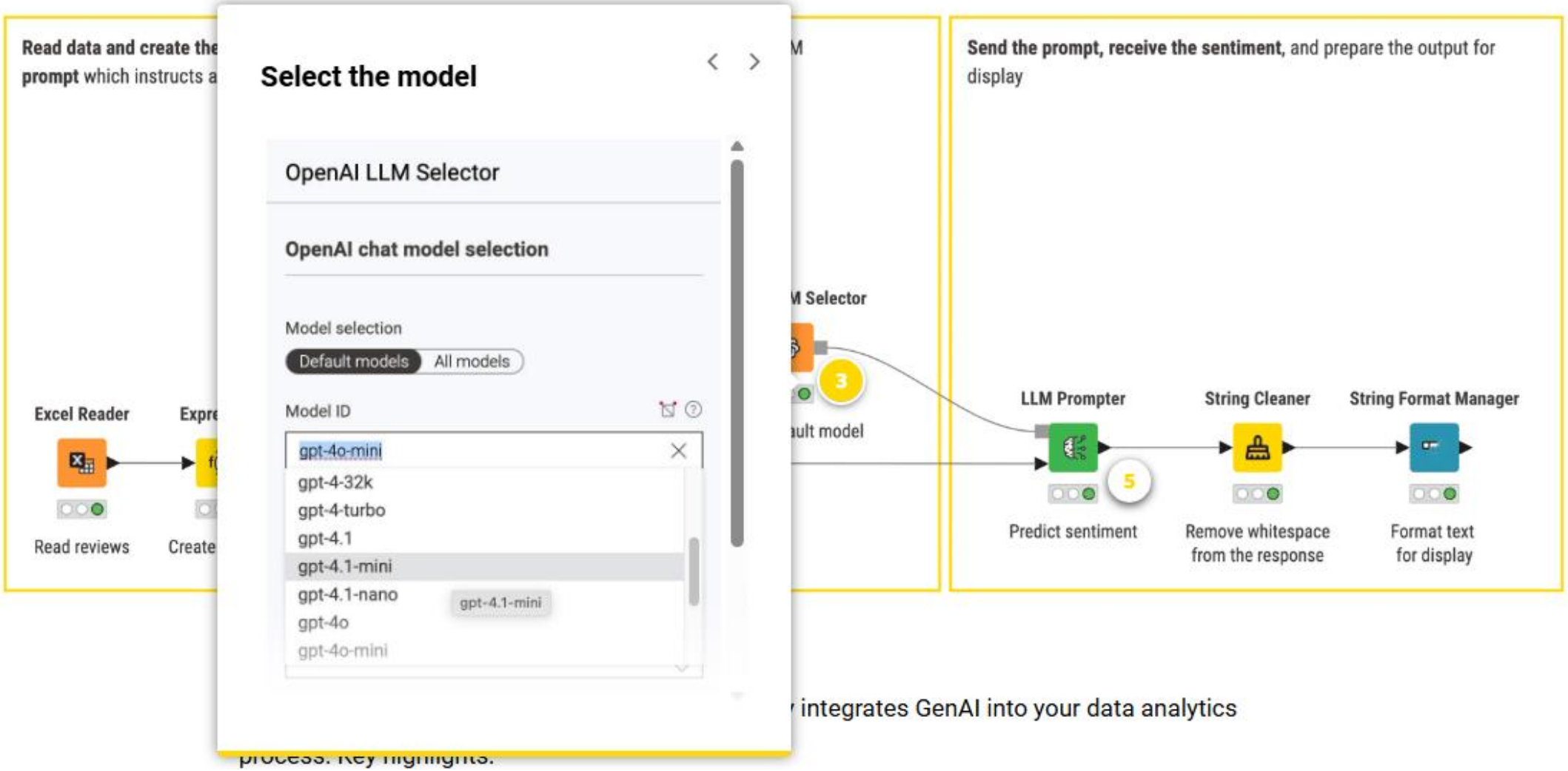


This simple sentir to your data analytics process. Key high

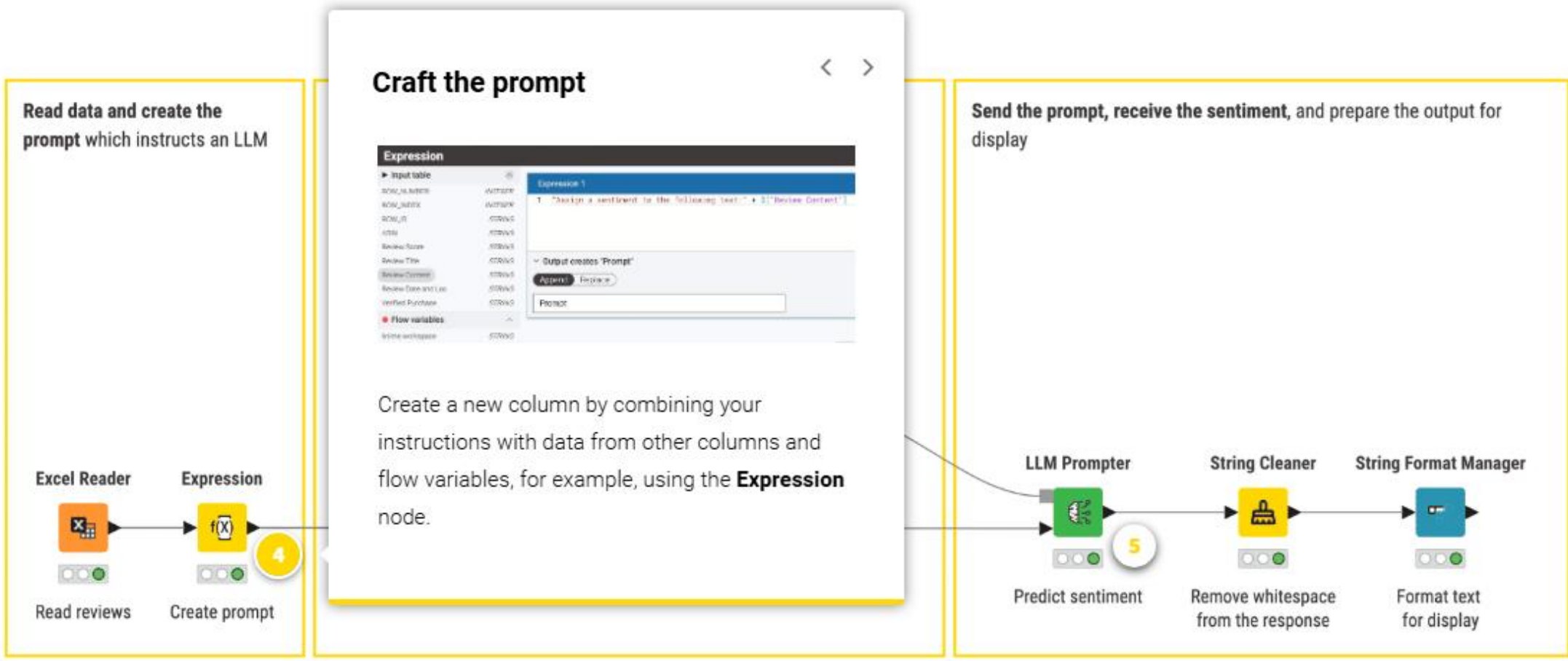
Authenticate and Connect



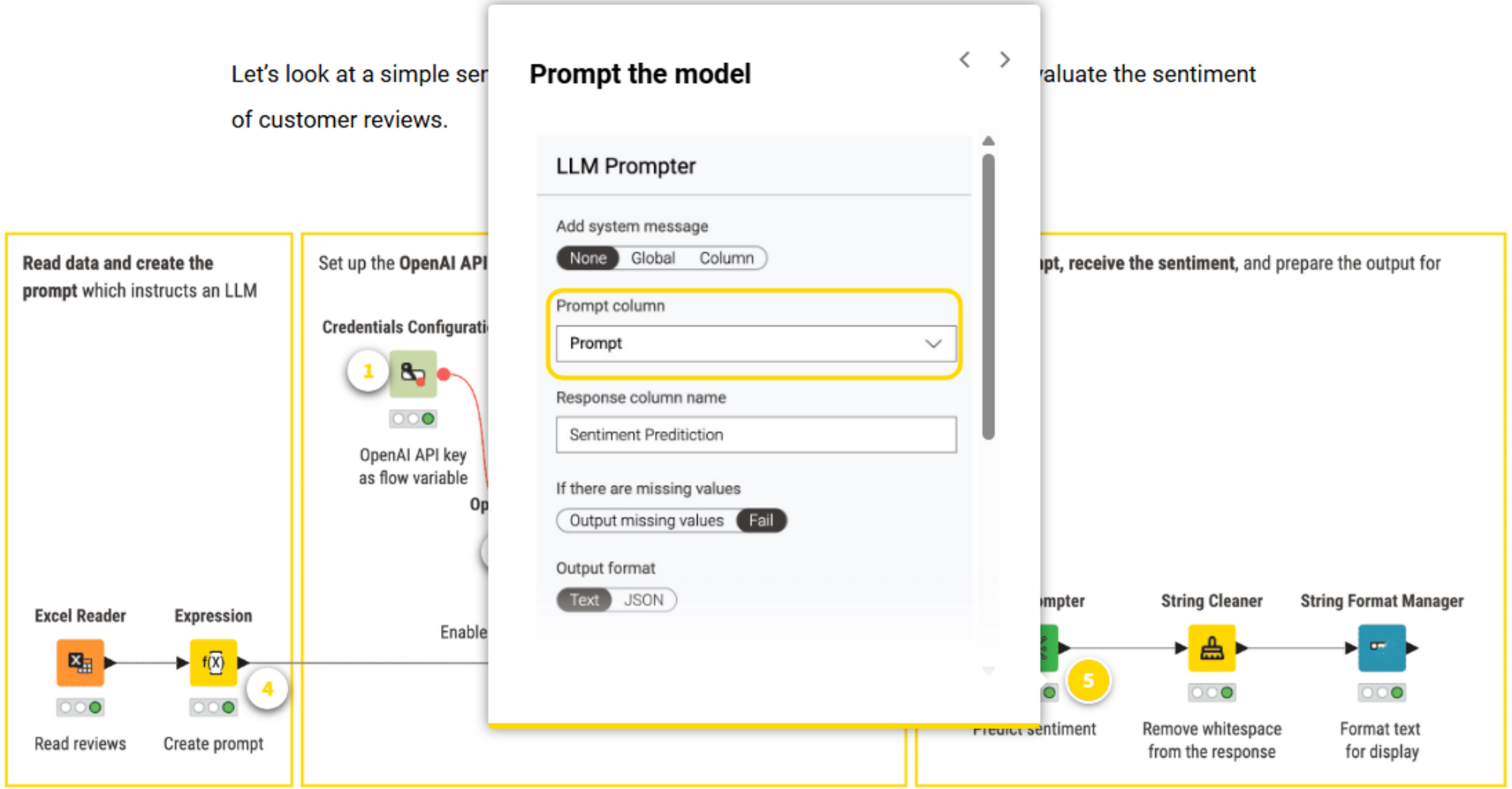
Select the Model



Craft the Prompt



Prompt the Model

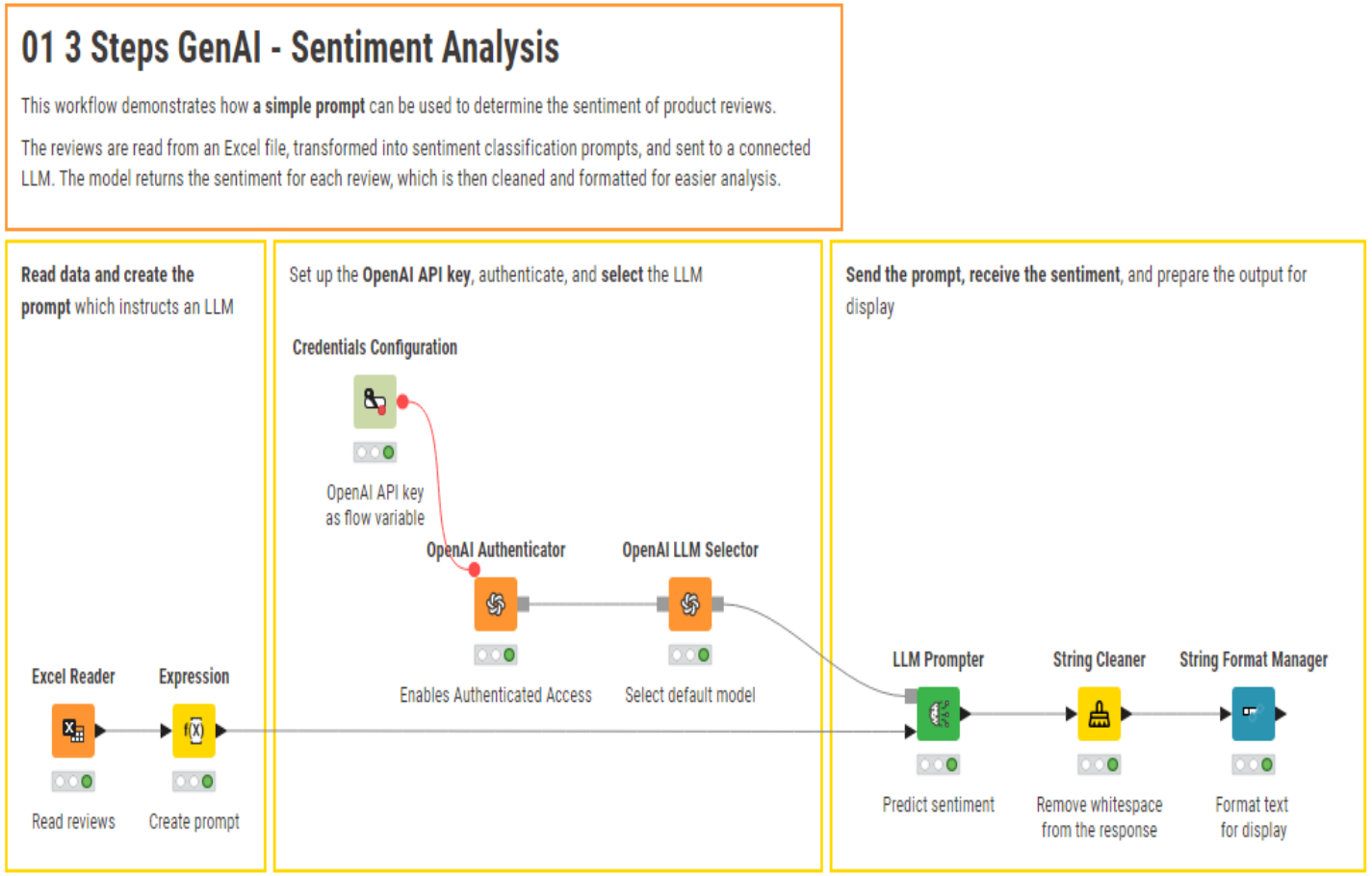


Sentiment Analysis Workflow

This simple sentiment analysis workflow integrates GenAI into data analytics process.

Key highlights:

- **Dynamic prompting**. Prompts are automatically tailored per row by embedding data directly into them. In the example, the instruction stays the same while the review text varies row by row.
- **Row-wise processing**. The LLM analyzes review in each prompt independently and returns the response separately for each row.
- **Seamless integration**. Once sentiment is assigned by the LLM, you can proceed with classic data analysis, e.g., visualizing sentiment in a bar chart or tracking trends over time.
- The LLM Prompter processes each prompt independently, row by row, making it well-suited for GenAI-powered data analytics.



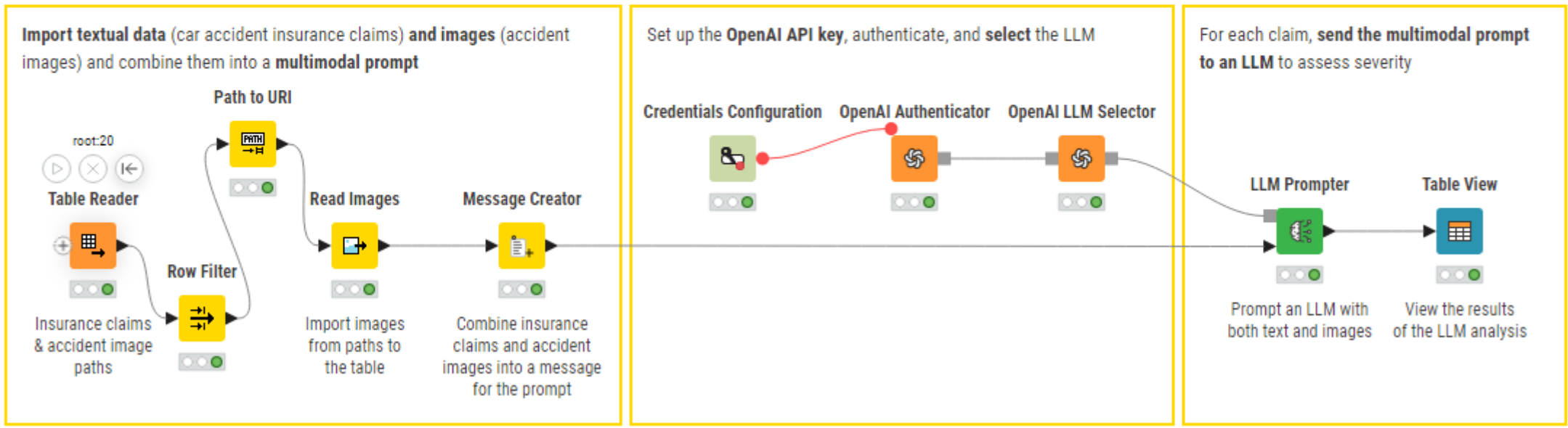
This workflow can be downloaded as following:

1. Download Course Workflows from VClass
2. Goto Generative AI Folder
3. Open 01 3 Steps GenAI workflow

02 Multimodal Prompting - Car Accident Severity Analysis

This workflow demonstrates how to **prompt an LLM using both text and images** to assess car accident severity.

Textual data from insurance claims and corresponding accident images are imported, combined into row-wise multimodal prompts, and sent to a connected LLM. The model then returns a severity assessment for each accident based on both the text and image inputs.



This workflow can be downloaded as following:

1. Download Course Workflows from VClass
2. Goto Generative AI Folder
3. Open 02 Multimodal Prompting

Local LLMs

Introduction to GPT4ALL

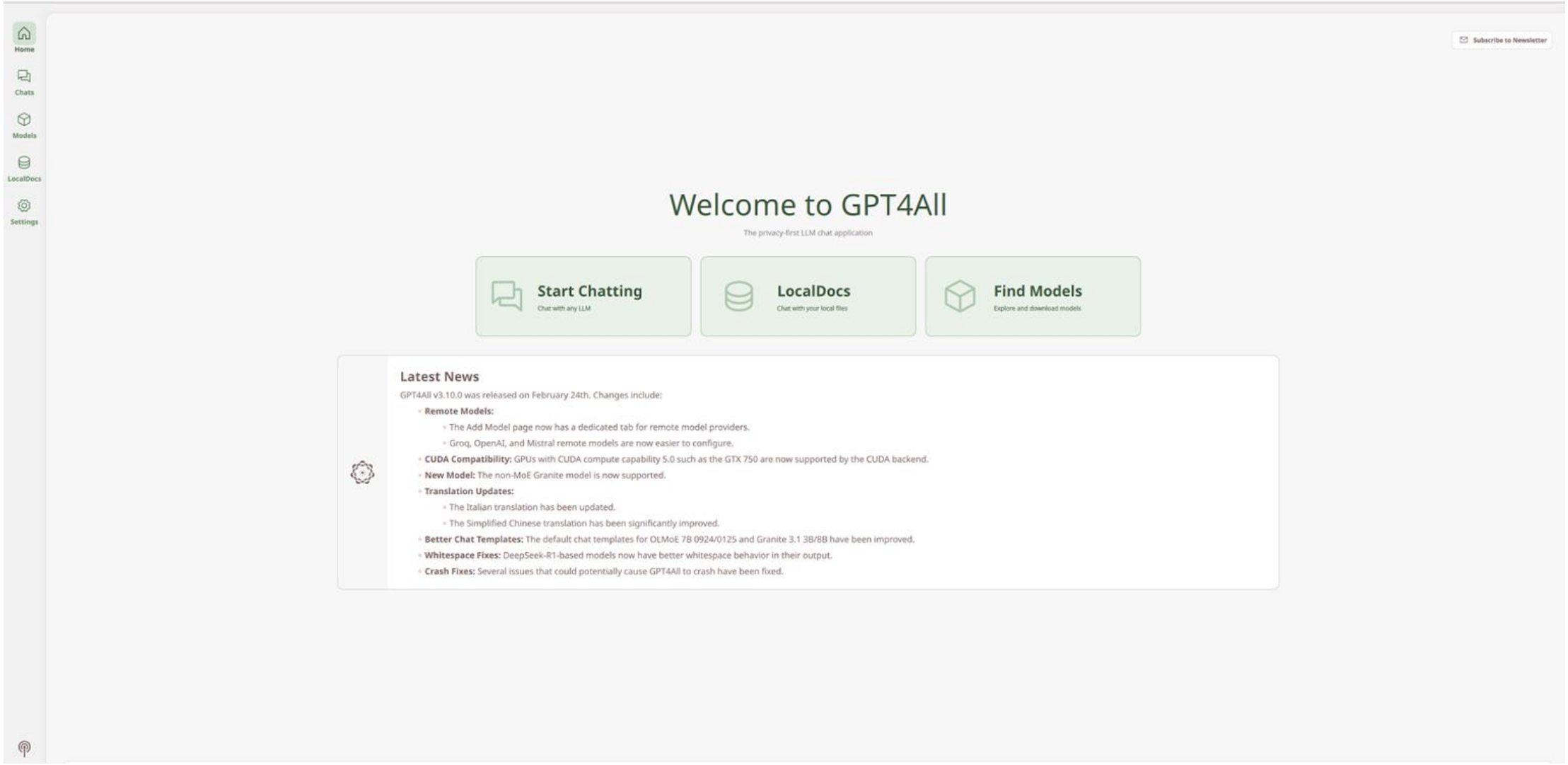
Introduction to GPT4All

- GPT4All is an open-source project from Nomic that makes it easy to run large language models entirely on your own computer.
- It bundles a cross-platform desktop chat app, a simple command-line tool, and Python bindings with a curated catalog of open-weight models.
- Those models are distributed in compact, quantized “.gguf” formats and run on lightweight runtimes (based on llama.cpp), so you can experiment on ordinary laptops without relying on cloud services or paying per token.

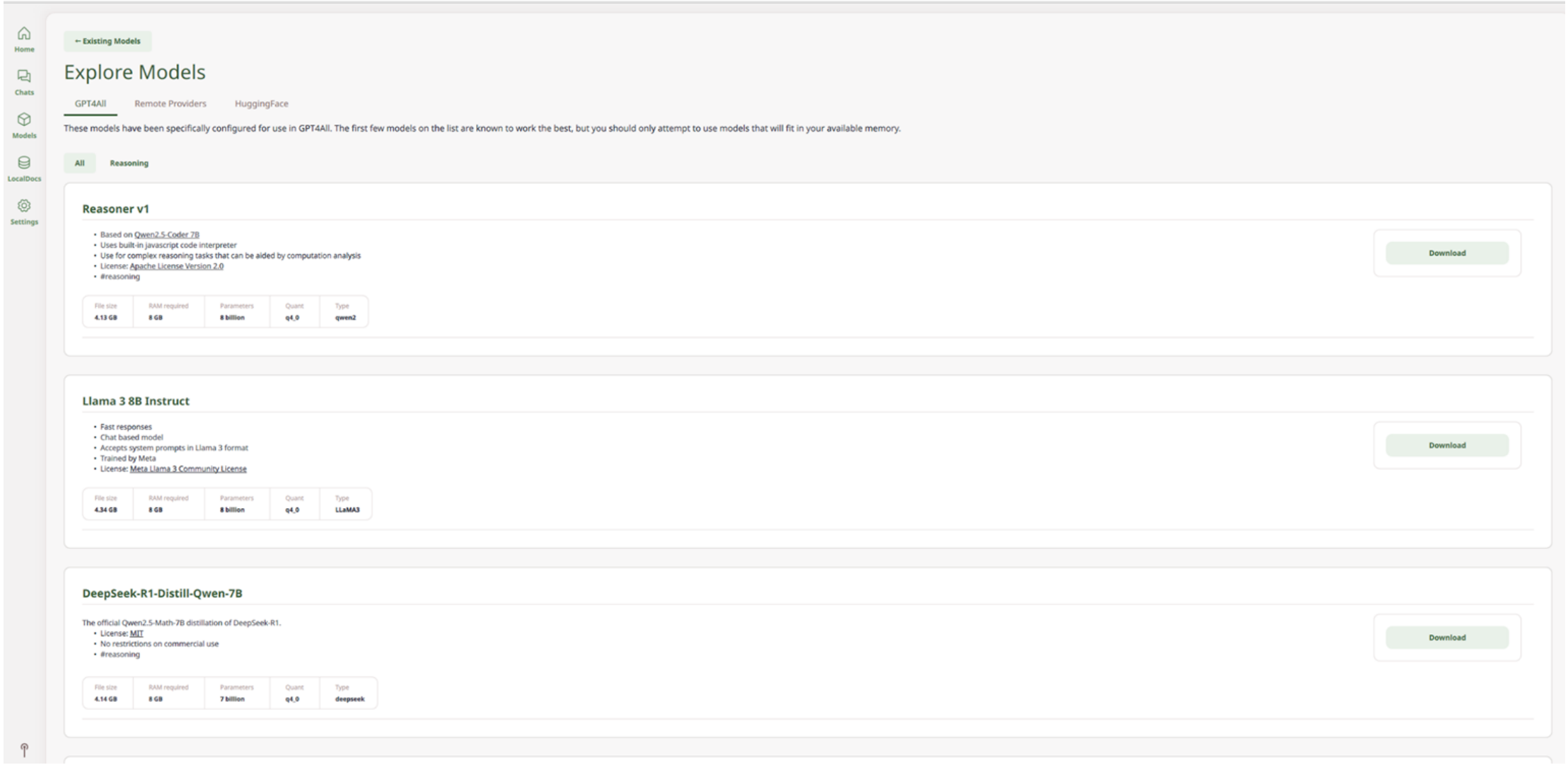
Introduction to GPT4All

- Using a **local LLM** might be a better choice:
  - **Data privacy**. Your data stays entirely within your system, which is critical when working with sensitive or confidential information.
  - **Offline availability**. Local models can run without internet access.
  - **Costs**. For organizations with high-volume usage, running models locally can reduce long-term costs, as there are no pay-per-use API fees.
- Everything runs locally, so prompts and outputs never leave your machine, there's no account or API key required, and you can work offline.
- You can choose from a range of models (instruction-tuned and chat variants of families like Llama, Mistral, Phi, and others), swap them in seconds, and tune generation settings such as temperature, top-p, context length, and max tokens.
- Performance depends on your hardware and the model size: smaller, more heavily quantized models are faster and lighter but less capable than larger ones.
- Typical uses include prototyping assistants, coding helpers, note summarizers, and RAG experiments where you combine a local model with your own documents.
- In tools like KNIME, GPT4All integrates through a local connector node, so you can drop it into a workflow, feed prompts from tables, and capture responses alongside the rest of your data pipeline.
- The main trade-offs are slower throughput than cloud GPUs, multi-gigabyte downloads for models, and quality that generally trails the newest frontier systems, but for many exploratory and privacy-sensitive tasks it offers a straightforward, zero-cost way to work with LLMs.

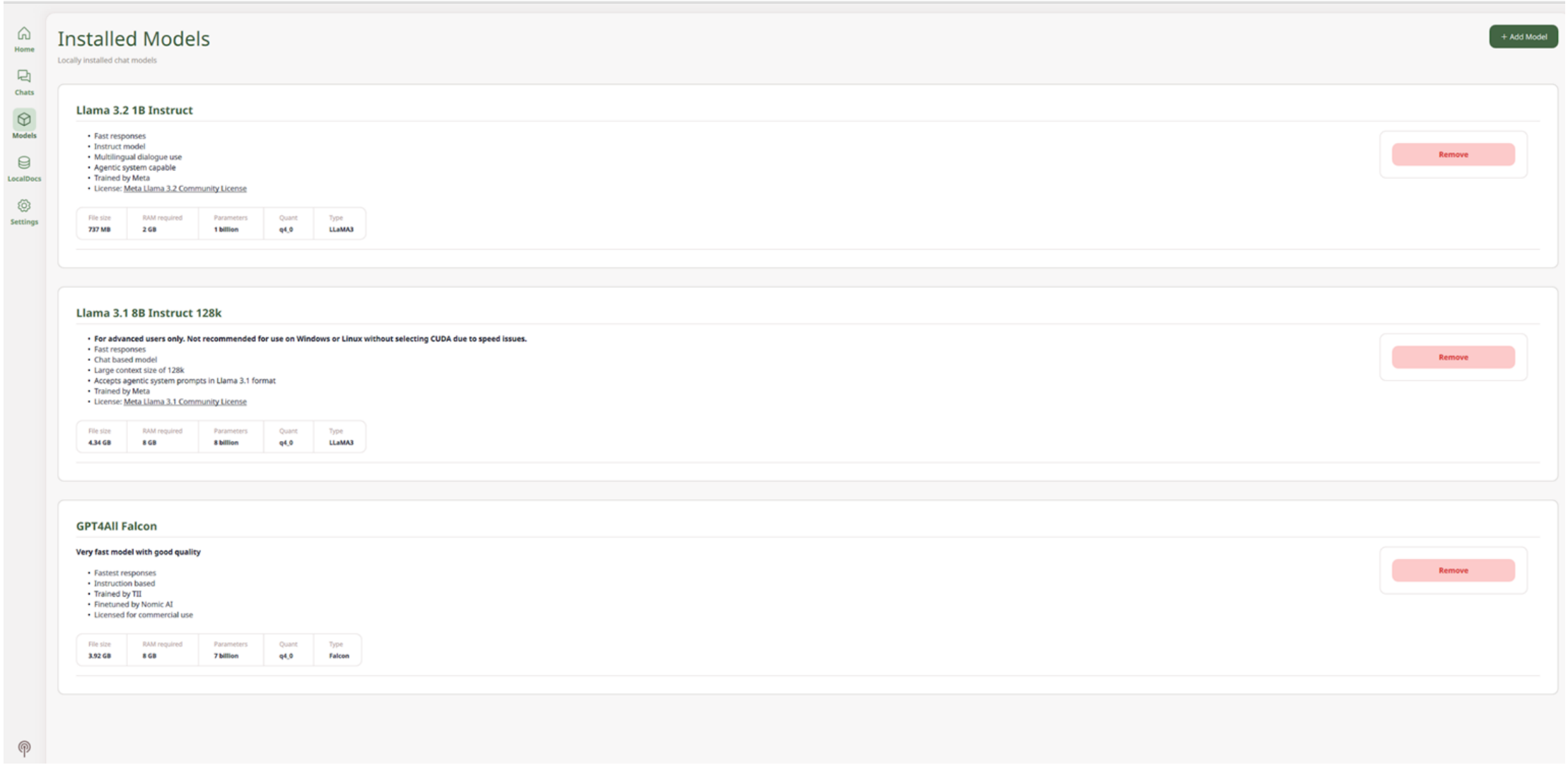
GPT4All - Home Screen



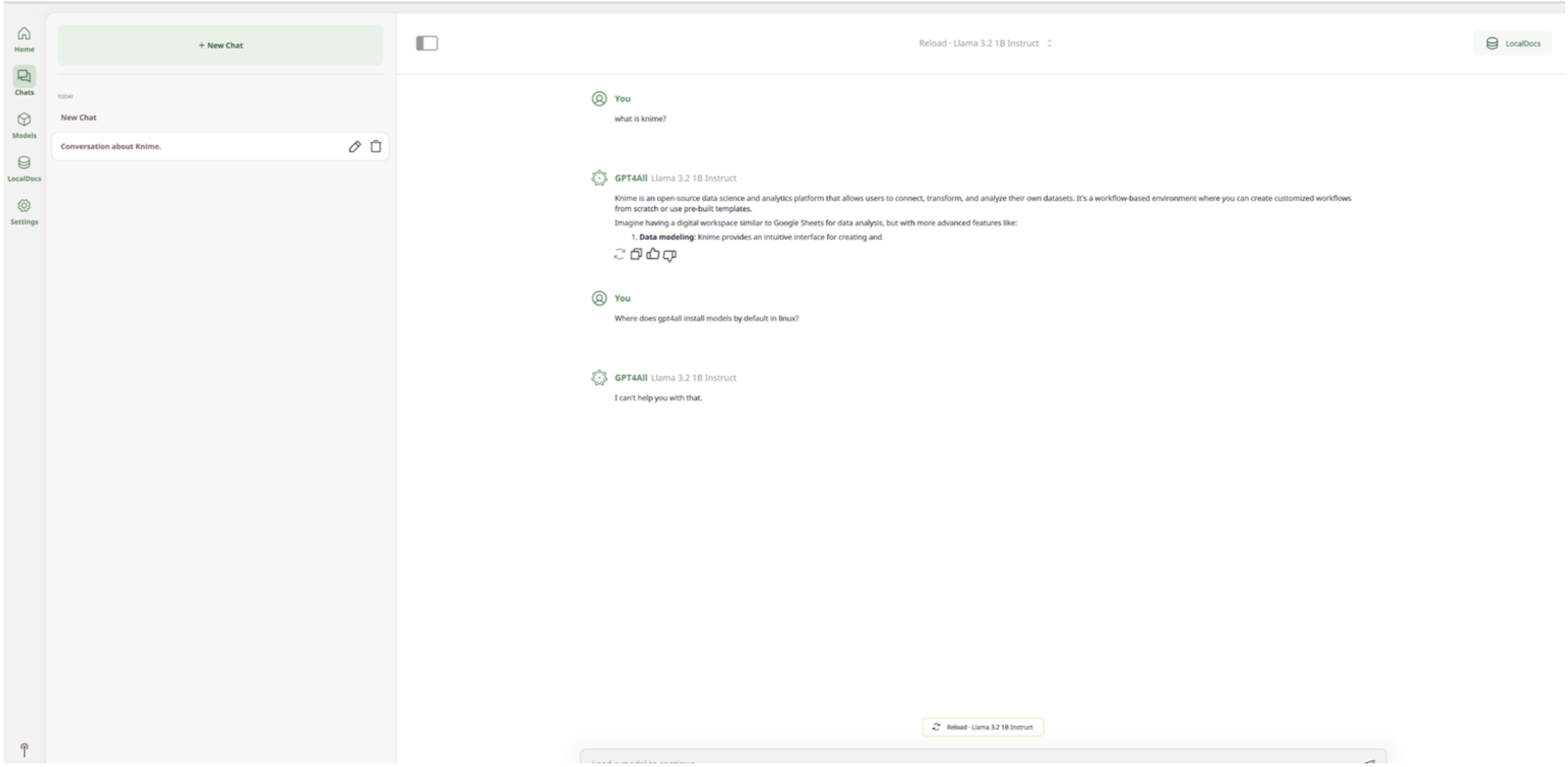
Explore and Download Models



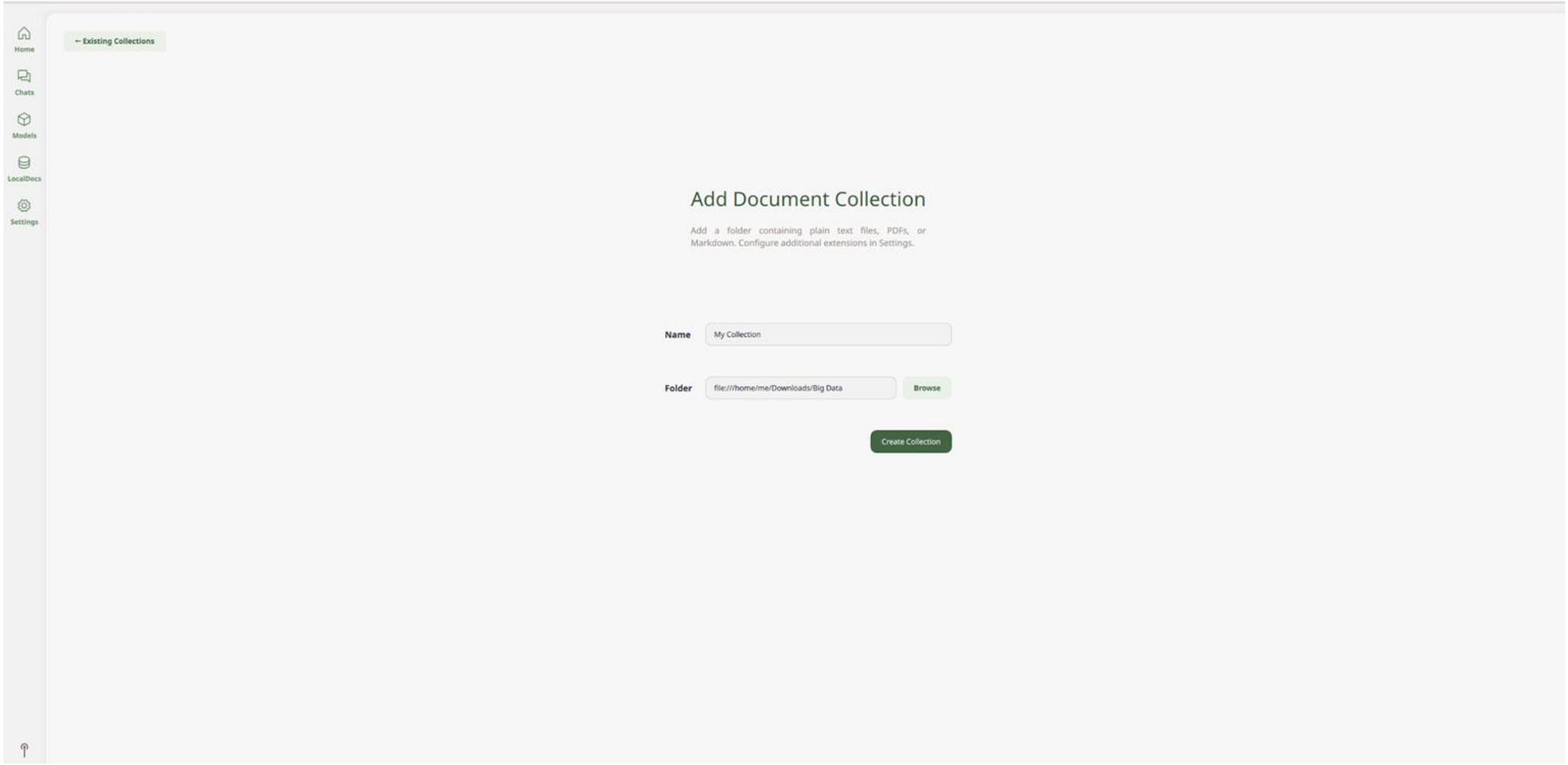
GPT4ALl – Installed Models



Chatting with Installed Models



Chat with your Own Documents



Step-by-step: KNIME with GPT4All (free, local)

Prerequisites

1. Install KNIME Analytics Platform.
2. Install the KNIME AI Extension: File → Install KNIME Extensions… → KNIME Labs → AI.
3. Ensure you have enough disk space (4–10 GB recommended) and RAM (8–16 GB recommended).

Download a GPT4All model

1. Install the GPT4All desktop app or visit the GPT4All model catalog.
2. Download a .gguf model suitable for CPU use. Choose a smaller, quantized file for speed and lower RAM, for example a Q4 or Q5 variant of an instruct model.
3. Note the full path to the downloaded .gguf file on your machine.

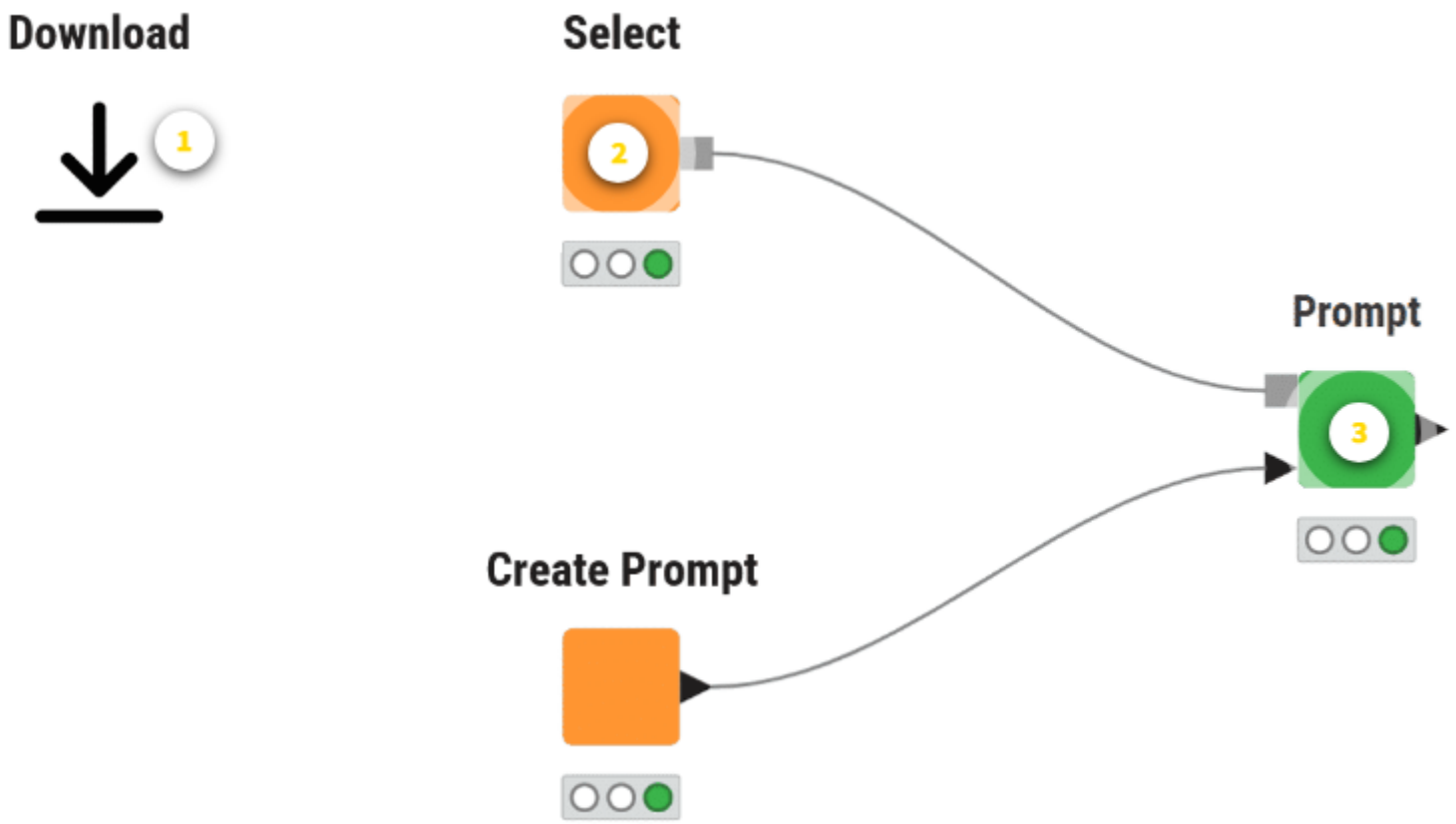
Build the KNIME workflow

1. Create a new workflow.
2. From the Node Repository, add: Local GPT4All LLM Selector, and either LLM Prompter or LLM Chat Prompter.
3. Optional: add a Table Creator or String Configuration node if you want to pass prompts from a table or a configuration dialog.

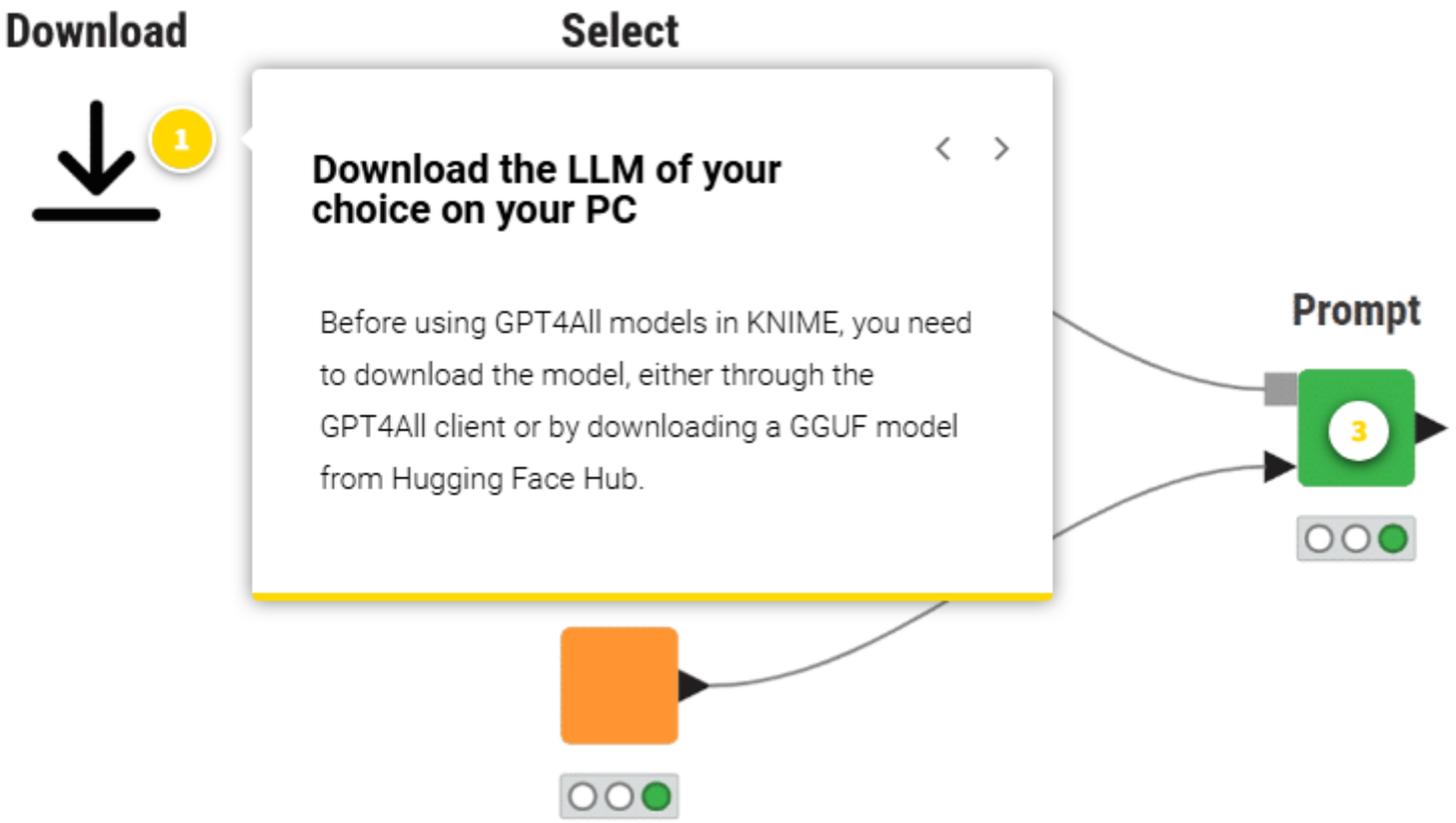
Configure Local GPT4All LLM Selector

1. Open the node configuration.
2. Model file: browse to your downloaded .gguf file.
3. Context length: start with 2048.
4. Threads: set to the number of physical CPU cores on your machine.
5. Sampling parameters: temperature 0.2–0.7, top_p 0.9 as a reasonable default.
6. Save and close.

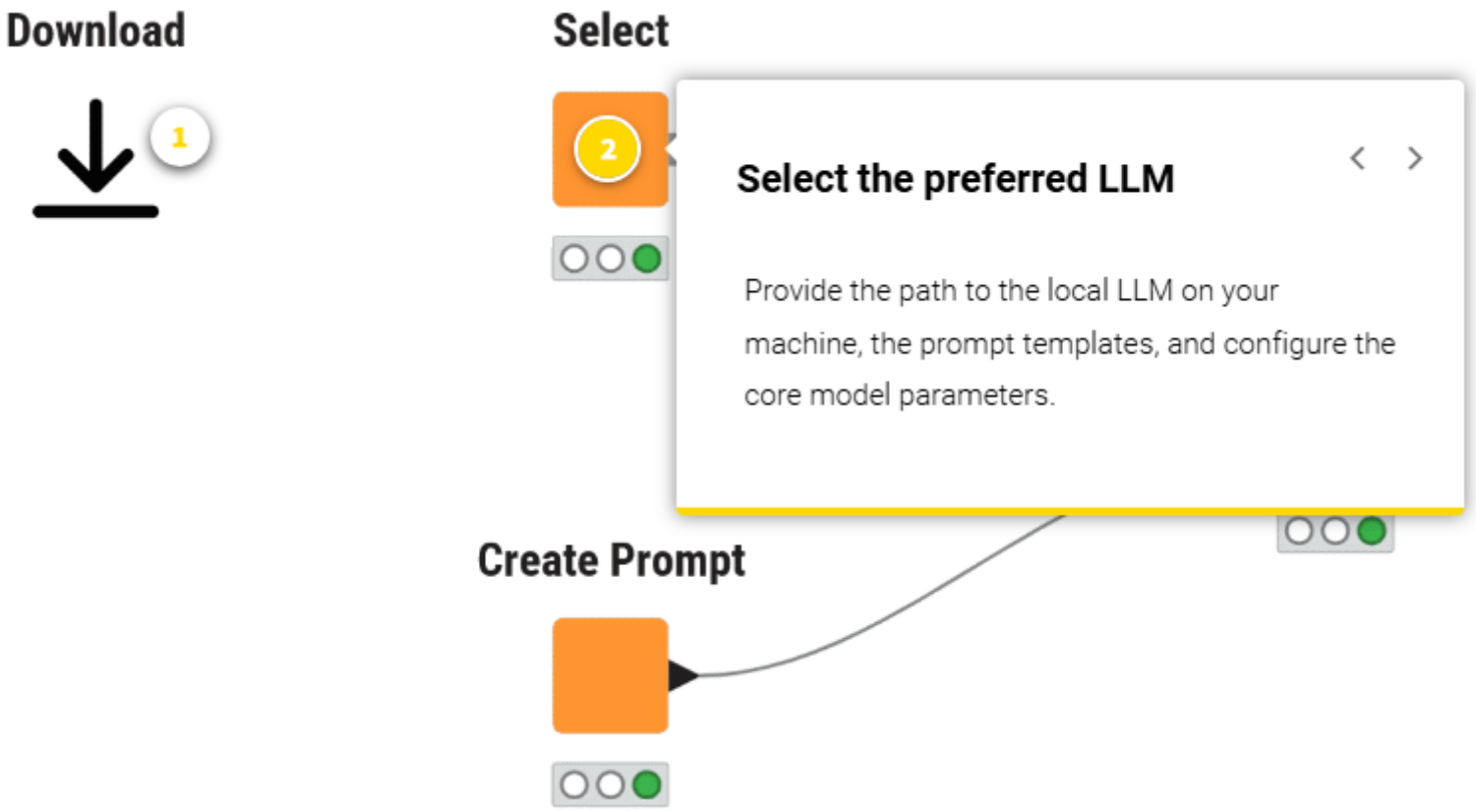
Working with Local LLMs Scenario



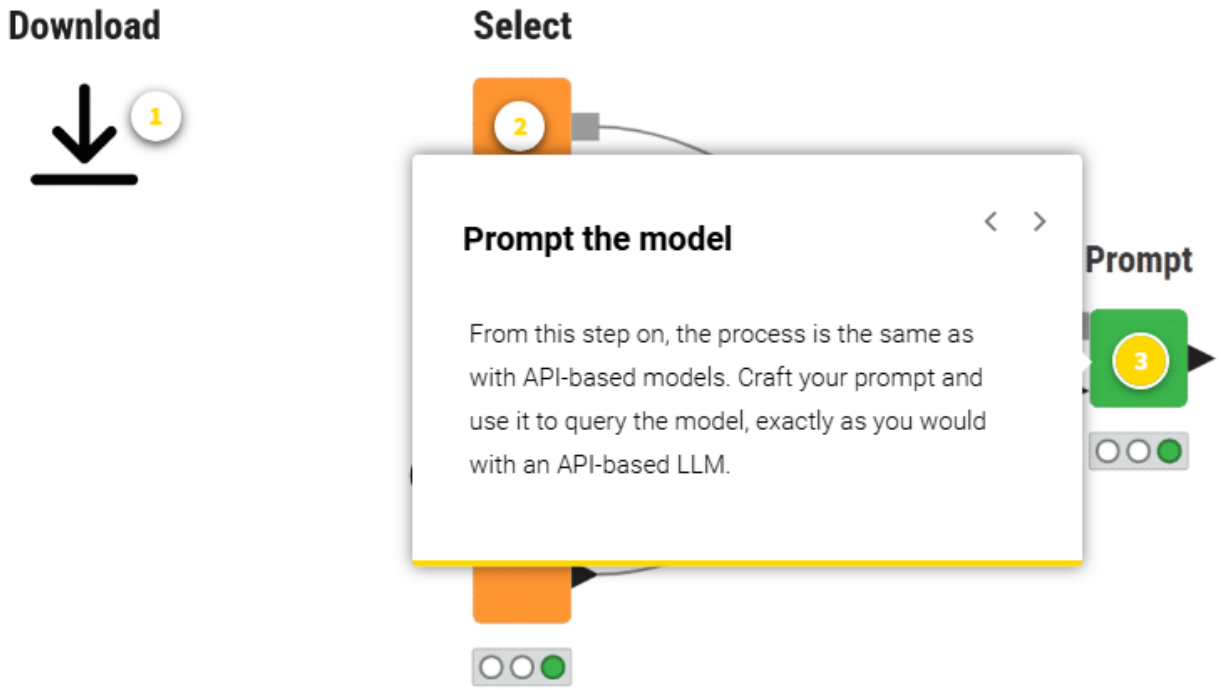
Download the LLM of your Choice



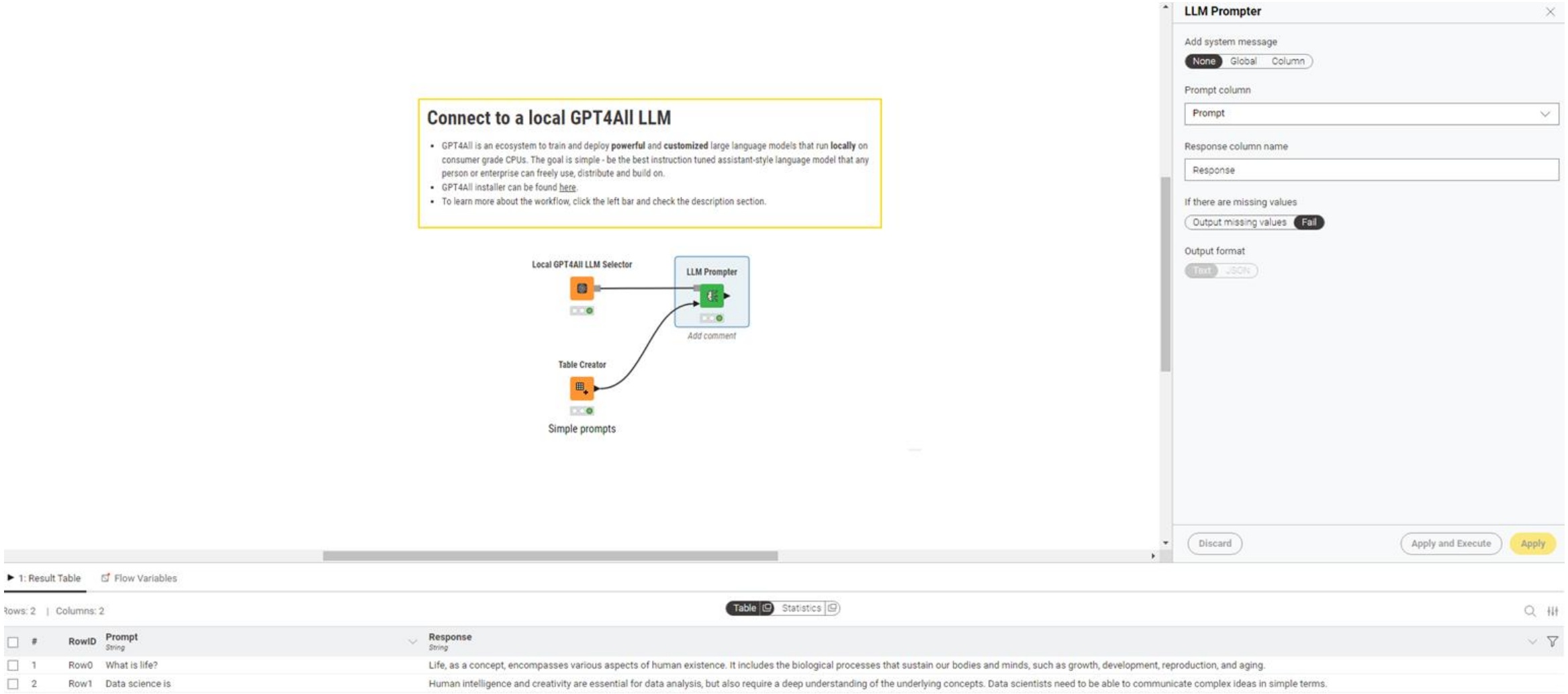
Set the Path to your LLM



Prompt the Model



You can craft your prompt as usual, without model-specific formatting. Your prompt will automatically be embedded into the template you provided in the model selector node.



This workflow can be downloaded as following:

1. Download Course Workflows from VClass
2. Goto Generative AI Folder
3. Open GPT4All Workflow